

ЛІТЕРАТУРА



НАВЧАЛЬНО-МЕТОДИЧНА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет
імені Івана Пулюя

Кафедра
комп'ютерних наук

**Методичні вказівки до виконання
лабораторних робіт
з дисципліни**

“ГРІД-СИСТЕМИ ТА ТЕХНОЛОГІЇ ХМАРНИХ ОБЧИСЛЕНЬ”

для студентів освітніх рівнів
«спеціаліст», «магістр»
122 «Комп'ютерні науки
та інформаційні технології»

**ТЕРНОПІЛЬ
2016**

УДК 681.3(07)
ББК 32.973-01
М54

Укладачі:

Шимчук Г.В., асистент,
Маєвський О.В., ст. викл.,
Назаревич О.Б., канд. техн. наук, ст. викл.,
Небесний Р.М., асистент.

Рецензент

М.М. Касянчук, канд. техн. наук, доцент.

Розглянуто й затверджено на засіданні
методичного семінару кафедри комп'ютерних наук
Тернопільського національного технічного університету імені Івана Пулюя.
Протокол № 2 від 09 вересня 2016р.

Схвалено та рекомендовано до друку на засіданні методичної комісії
факультету комп'ютерно-інформаційних систем та програмної інженерії
Тернопільського національного технічного університету імені Івана Пулюя.
Протокол № 2 від 23 вересня 2016р.

Методичні вказівки до виконання лабораторних робіт з дисципліни
К65 «Грид-системи та технології хмарних обчислень» для студентів освітніх
рівнів «спеціаліст» «магістр» / Укладачі : Шимчук Г.В., Маєвський О.В.,
Назаревич О.Б., Небесний Р.М. – Тернопіль : Тернопільський національний
технічний університет імені Івана Пулюя , 2016. – 52 с.

УДК 681.3(07)
ББК 32.973-01

Методичні вказівки до виконання лабораторних робіт призначені для
полегшення засвоєння дисципліни “Грид-системи та технології хмарних обчислень”.
Складається з урахуванням модульної системи навчання.

Відповідальний за випуск *М.В. Приймак*, докт. техн. наук, професор.

© Шимчук Г.В., Маєвський О.В.,
Назаревич О.Б., Стадник М.А.,..... 2016
Тернопільський національний технічний
© університет імені Івана Пулюя,..... 2016

ЗМІСТ

ЛАБОРАТОРНА РОБОТА № 1.

| | |
|----------------------------------|---|
| ХМАРА. ХМАРНЕ СХОВИЩЕ ДАНИХ..... | 4 |
|----------------------------------|---|

ЛАБОРАТОРНА РОБОТА № 2.

| | |
|--|----|
| СТВОРЕННЯ ВЕБ-ДОДАТКУ WORD PRESS ЗА ДОПОМОГОЮ ХМАРНОГО СЕРВІСУ WINDOWS AZURE..... | 13 |
|--|----|

ЛАБОРАТОРНА РОБОТА № 3.

| | |
|---|----|
| РОЗГОРТАННЯ ВЕБ-САЙТУ НА AZURE ЗА ДОПОМОГОЮ MICROSOFT VISUAL STUDIO..... | 36 |
|---|----|

ЛАБОРАТОРНА РОБОТА № 1

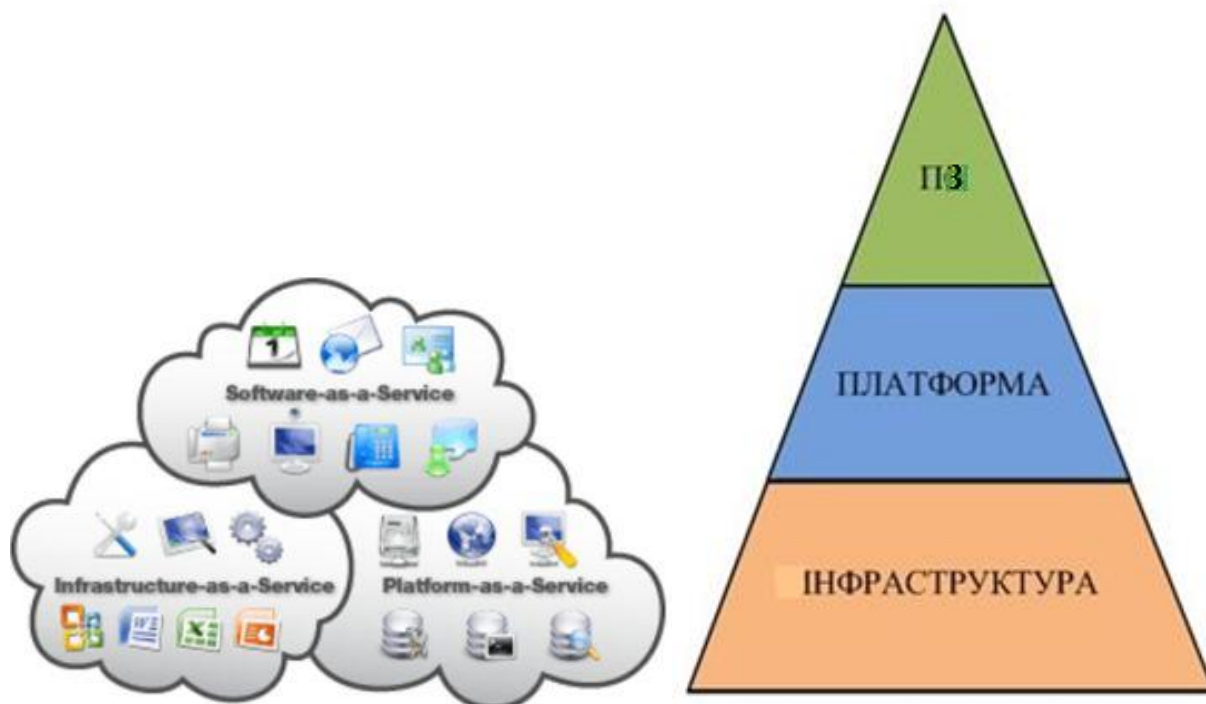
ТЕМА: ХМАРА. ХМАРНЕ СХОВИЩЕ ДАНИХ

Реєстрація на Microsoft Azure

«Хмара» – це технологія розподіленої обробки даних в якій комп'ютерні ресурси і потужності надаються користувачеві як Інтернет-сервіс, тобто робочий майданчик на віддаленому сервері.

Різниця полягає виключно в методі зберігання і обробки даних. Якщо всі операції відбуваються на комп'ютері користувача (з використанням його потужностей), то це – не «хмара», а якщо процес відбувається на сервері в мережі, то це «хмарні технології» – різні апаратні, програмні засоби, методології та інструменти, що надаються користувачеві, як Інтернет-сервіси, для реалізації своїх цілей, завдань, проектів.

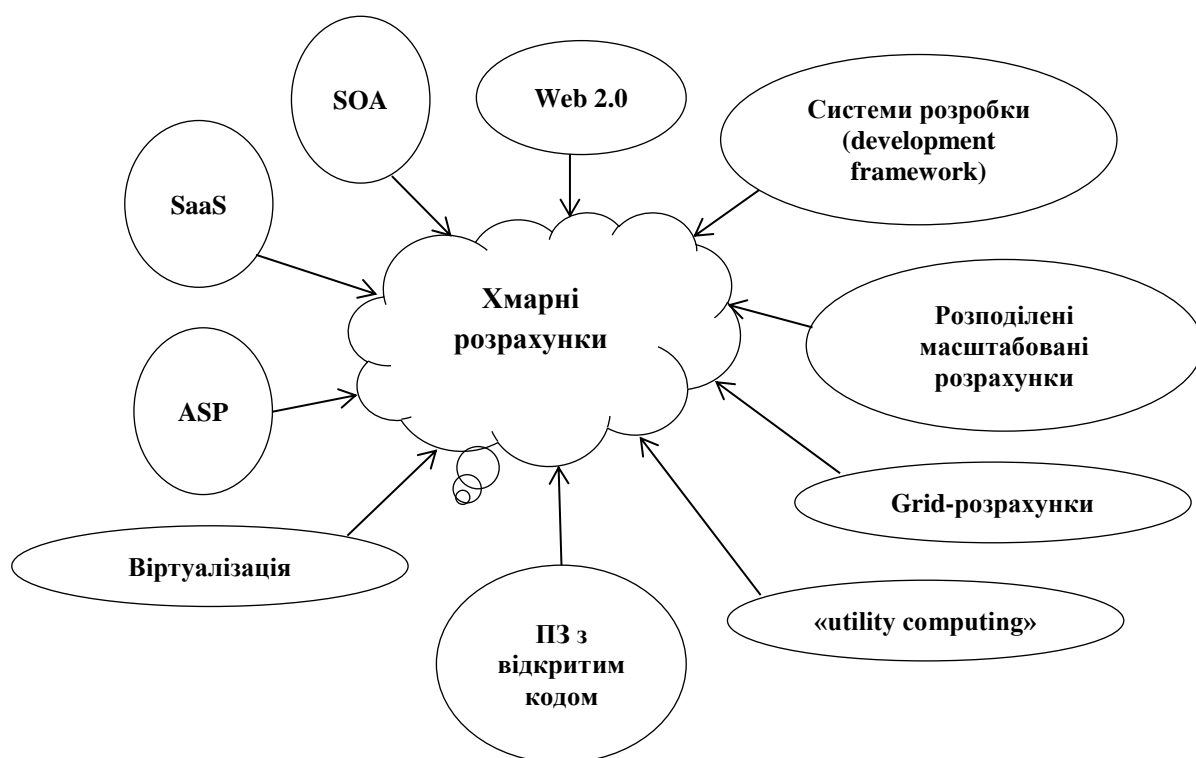
Терміни «хмарні технології»/«хмарний сервіс», з їх загальноприйнятим графічним представленням, у вигляді «хмарок», тільки плутає користувачів, насправді їх структуру, можна легко зрозуміти, якщо уявити її у вигляді такої піраміди.



Основа піраміди «інфраструктура» – це набір фізичних пристроїв (сервери, тверді диски), над нею надбудовується «платформа» – набір послуг і верхівка – програмне забезпечення, що доступне за запитом користувачів.

Хмарні обчислення – це певний базис-вектор, отриманий в результаті синтезу цілого ряду технологій і підходів.

Хмарні технології – це набір засобів, що виконує обчислення за допомогою віддалених серверів і програм без безпосереднього залучення ресурсів комп'ютера користувача.



Хмарне сховище даних – модель онлайн-сховища, в якому дані зберігаються на численних, розподілених в мережі серверах, що надаються в користування клієнтам, в основному третьою стороною. На противагу моделі зберігання даних на власних, виділених серверах, що придбавалися або орендованих спеціально для подібних цілей, кількість або яка-небудь внутрішня структура серверів клієнтові, в загальному випадку, не видна.

Дані зберігаються, а рівно і обробляються, в так званій хмарі, яка є, з точки зору клієнта, один великий, віртуальний сервер.

Категорії «хмар» за формою власності:

Публічна хмара – це IT-інфраструктура, яка використовується одночасно багатьма компаніями і сервісами. Користувачі не мають можливості управляти і обслуговувати дану «хмару», відповідальність з цих питань покладено на власника ресурсу. Абонентом, пропонованих сервісів може стати будь-яка компанія та індивідуальний користувач.

Приватна хмара – це безпечна IT-інфраструктура, що контролюється і експлуатується в інтересах однієї організації. Організація може керувати приватною «хмарою» самостійно або доручити це завдання зовнішньому підряднику. Інфраструктура може розміщуватися або в приміщеннях замовника, або у зовнішнього оператора (або частково у замовника і частково у оператора).

Гібридна хмара – це IT-інфраструктура, що використовує найкращі якості публічної і приватної хмари при вирішенні поставленого завдання. Часто такий тип застосовується, коли організація має сезонні періоди активності, тобто, як тільки внутрішня IT-інфраструктура не справляється з поточними завданнями, частина потужностей перекидається на публічну «хмару» (наприклад, великі обсяги статистичної інформації), а також для надання доступу користувачам до ресурсів підприємства через публічну «хмару».

Послуги, що надаються хмарними системами:

Storage-as-a-Service (SaaS)

Найпростіший з CC-сервісів, що представляє собою дисковий простір на вимогу. Послуга Storage-as-a-Service дає можливість зберігати дані в зовнішньому сховищі, в «хмарі». Для користувача воно буде виглядати, як додатковий логічний диск або папка. Сервіс є базовим для інших, оскільки входить до складу практично кожного з них.

Database-as-a-Service (DaaS)

Надає можливість працювати з базами даних, подібно так, як СУБД було встановлено на локальному ресурсі. В цьому випадку значно легше розділяти проекти між різними виконавцями та заощадити на комп'ютерному обладнанні та ліцензіях, необхідних для грамотного використання СУБД в великій чи середньої організації.

Platform-as-a-Service (PaaS)

Користувачеві надається комп'ютерна платформа з встановленою операційною системою і певним програмним забезпеченням.

Security-as-a-Service (SaaS)

Даний вид послуги надає можливість користувачам швидко розгортати продукти, що вимагають безпечне використання веб-технологій, електронного листування, локальної мережі. Користувачі даного сервісу мають змогу економити на розгортанні та підтримці своєї власної системи безпеки.

Infrastructure-as-a-Service (IaaS)

Користувачеві надається комп'ютерна інфраструктура, зазвичай віртуальні платформи (комп'ютери), пов'язані в мережу, які він самостійно налаштовує під власні цілі.

Testing-as-a-Service (TaaS)

Надає можливість тестування локальних або «хмарних» систем з використанням тестового ПЗ з «хмари» (при цьому жодного устаткування або забезпечення на підприємстві, не потрібно).

Порядок виконання лабораторної роботи

1. Заходимо по зсилці <https://www.dreamspark.com/>

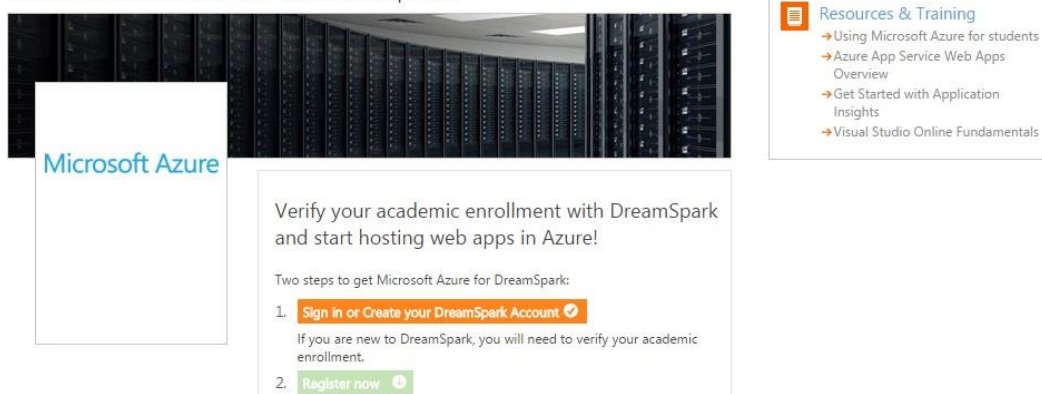


2. Вибираємо **Get Azure now** →

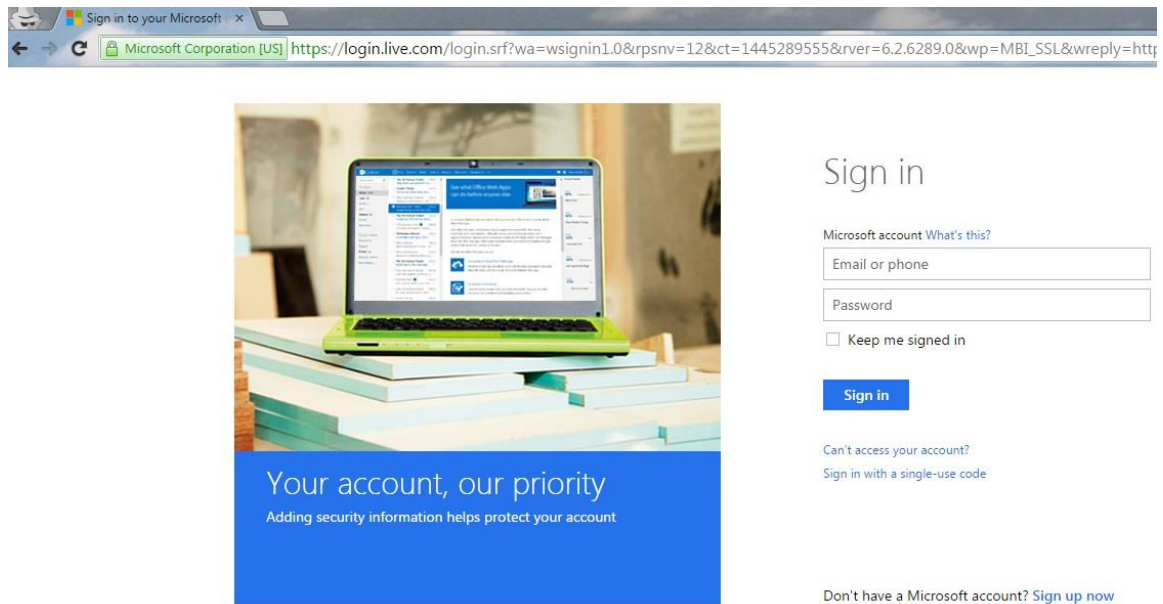


3. На **Microsoft Azure for DreamSpark** створюємо аккаунта, нажимаючи на **“Sign in or Create your DreamSpark Account”**

Microsoft Azure for DreamSpark



4. Далі заходимо під акаунтом Microsoft



Якщо аккаунта не маєте реєструємося

Don't have a Microsoft account? [Sign up now](#)

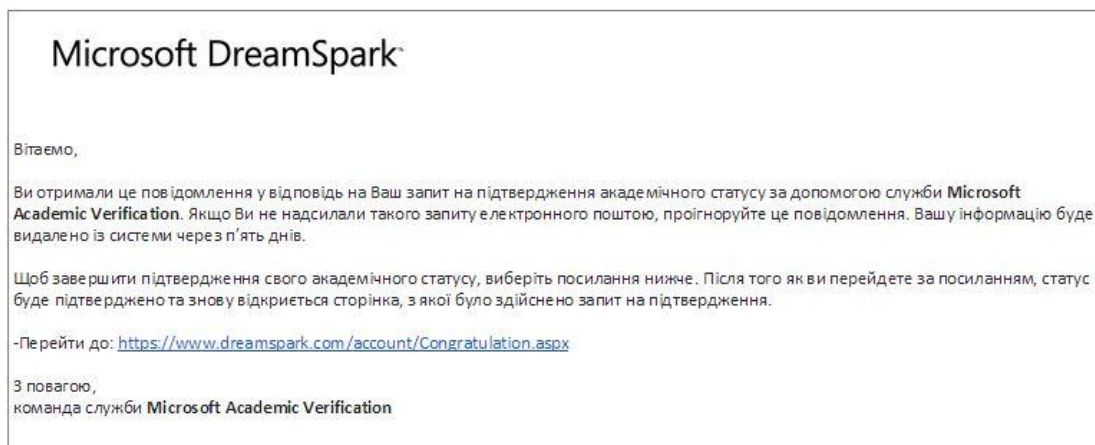
5. Після входу під акаунтом Microsoft, переходимо для реєстрації. Методом перевірки, буде представлення студентського квитка. Після заповнення усіх полів, відправляємо дані.

Во всех разделах пользовательского интерфейса, заданных для подтверждения академического статуса, под учебным заведением подразумевается школа, университет, колледж, техническое училище, школа, специализирующаяся на естественных, технических, инженерных науках или математике, или другое образовательное учреждение.

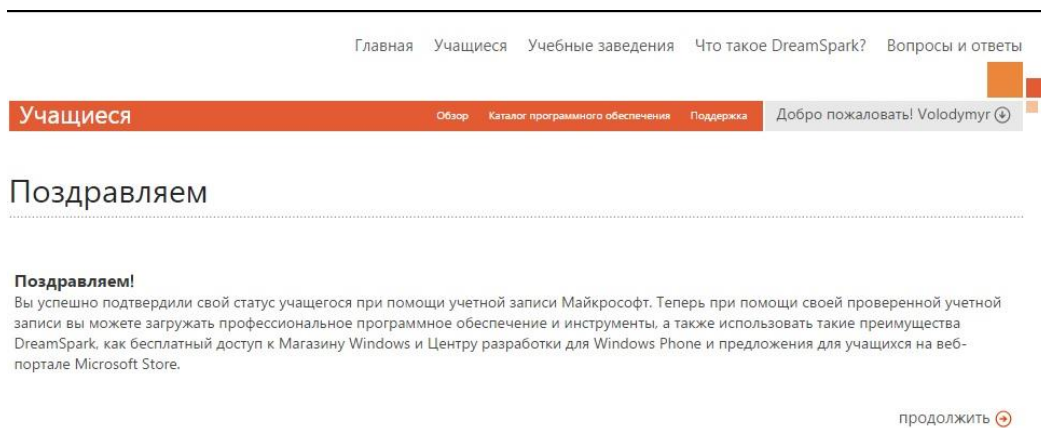
Сначала выберите метод проверки.

| | | | | | | | | | | | | | | | | | |
|---|---|----------------------|------------------------------------|--------------------------------------|----------------------|---------------------------|---------------|---|--|--------------------------------------|------------------------------|---|----------------------|--------------|--|----------------------|--|
| <div>У меня есть учетная запись электронной почты учебного заведения</div> <div>У меня есть учетная запись учебного заведения</div> <div>У меня есть международная идентификационная карта студента</div> <div>Мне выдали код проверки</div> <div>Я могу предоставить документы</div> | <p>Я могу предоставить документы</p> <p>Допустимые документы: студенческий билет с датой, текущий отчет о выполнении работ, текущее расписание занятий с датой или извещение о приеме в высшее учебное заведение.</p> <table border="0"><tr><td>* страна или регион:</td><td>* Имя подтверждающего документа: ?</td></tr><tr><td><input type="text" value="Украина"/></td><td><input type="text"/></td></tr><tr><td>* Тип учебного заведения:</td><td>* Вложение: ?</td></tr><tr><td><input type="text" value="Высшее образование"/></td><td><input type="text" value="Student.jpg"/> <input data-bbox="1203 1776 1289 1809" type="button" value="Обзор..."/></td></tr><tr><td>* Наименование учебного заведения: ?</td><td>* Адрес электронной почты: ?</td></tr><tr><td><input type="text" value="Тернопільський Національний Технічний Унів"/></td><td><input type="text"/></td></tr><tr><td colspan="2">Комментарий:</td></tr><tr><td colspan="2"><input type="text"/></td></tr></table> | * страна или регион: | * Имя подтверждающего документа: ? | <input type="text" value="Украина"/> | <input type="text"/> | * Тип учебного заведения: | * Вложение: ? | <input type="text" value="Высшее образование"/> | <input type="text" value="Student.jpg"/> <input data-bbox="1203 1776 1289 1809" type="button" value="Обзор..."/> | * Наименование учебного заведения: ? | * Адрес электронной почты: ? | <input type="text" value="Тернопільський Національний Технічний Унів"/> | <input type="text"/> | Комментарий: | | <input type="text"/> | |
| * страна или регион: | * Имя подтверждающего документа: ? | | | | | | | | | | | | | | | | |
| <input type="text" value="Украина"/> | <input type="text"/> | | | | | | | | | | | | | | | | |
| * Тип учебного заведения: | * Вложение: ? | | | | | | | | | | | | | | | | |
| <input type="text" value="Высшее образование"/> | <input type="text" value="Student.jpg"/> <input data-bbox="1203 1776 1289 1809" type="button" value="Обзор..."/> | | | | | | | | | | | | | | | | |
| * Наименование учебного заведения: ? | * Адрес электронной почты: ? | | | | | | | | | | | | | | | | |
| <input type="text" value="Тернопільський Національний Технічний Унів"/> | <input type="text"/> | | | | | | | | | | | | | | | | |
| Комментарий: | | | | | | | | | | | | | | | | | |
| <input type="text"/> | | | | | | | | | | | | | | | | | |

6. Очікуємо 1-2 дні і перевіряємо пошту.

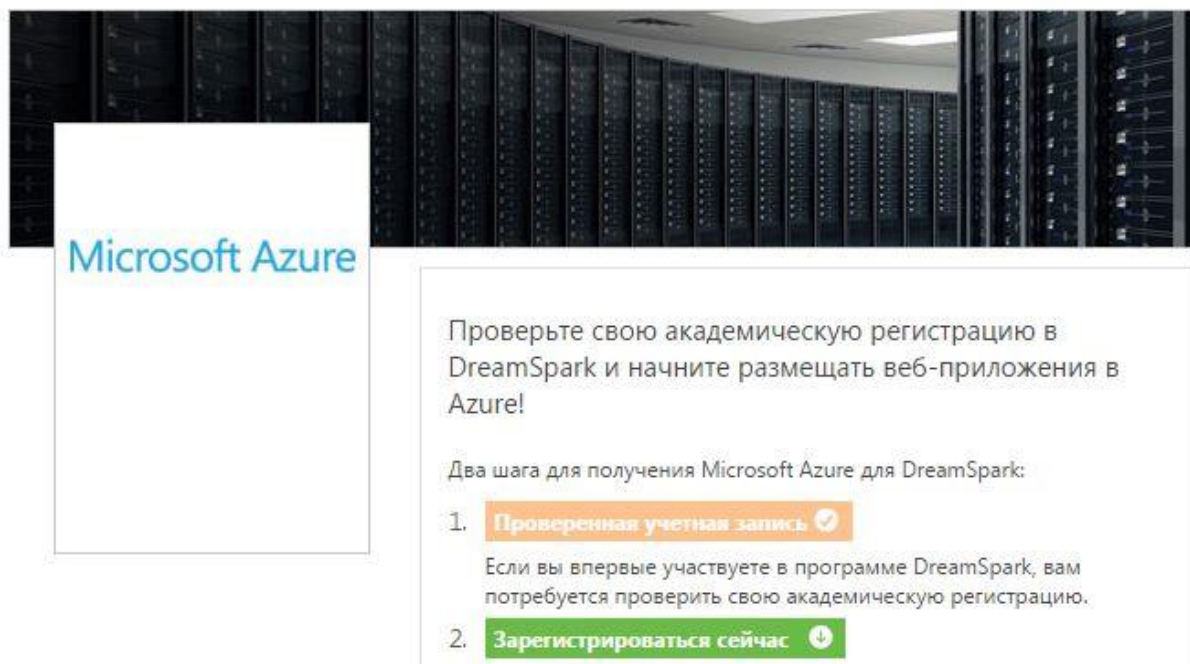


Переходимо по зсилці і підтверджуємо. Нажимаємо «продолжить»



7. На даній сторінці вибираємо «Зарегистрироваться сейчас»

Microsoft Azure для DreamSpark



8. Заповнюємо поля, які є обов'язковими і вводимо номер телефона свій, в даному випадку на телефон прийде SMS з кодом для введення.

Регистрация

DreamSpark

Дополнительные сведения ▼

Microsoft Azure

1 О вас

ИМЯ ФАМИЛИЯ СТРАНА/РЕГИОН ?

АДРЕС ЭЛ. ПОЧТЫ ? КОМПАНИЯ ИЛИ УЧЕБНОЕ ЗАВЕДЕНИЕ РАБОЧИЙ ТЕЛЕФОН

ЗАРЕГЕСТРИРОВАННОСТЬ

2 Проверка по телефону ?

☒ Отправить текстовое сообщение ☐ Позвонить мне

Россия (+7)

Отправить текстовое сообщение

3 Соглашение

☐ Я принимаю соглашение на подписку, сведения о соглашении и заявление о конфиденциальности.

☐ Корпорация Майкрософт может использовать мой адрес электронной почты и номер телефона для передачи информации о специальных предложениях, относящихся к Microsoft Azure.

9. Завершения підготовки

Подождите, пока выполняется подготовка.

Обычно это занимает менее 4 мин. По завершении вы получите электронное письмо.



Ваша підписка на Microsoft Azure готова

Ваша подписка готова.

Начать управлять моей службой >

Запитання до лабораторної

1. Що таке хмара?
2. Що таке хмарні технології?
3. Що таке хмарне сховище даних?
4. Категорії хмар за формою власності?
5. Infrastructure-as-a-Service це?

Список використаної літератури:

1. <http://victoria.lviv.ua/html/webproject/doc/1.3.doc>
2. John W. Rittinghouse, James F. Ransome – «Cloud Computing: Implementation, Management, and Security.

ЛАБОРАТОРНА РОБОТА № 2

ТЕМА: СТВОРЕННЯ ВЕБ-ДОДАТКУ WORD PRESS ЗА ДОПОМОГОЮ ХМАРНОГО СЕРВІСУ WINDOWS AZURE

Відомості про платформу Windows Azure

Microsoft Azure – операційна система корпорації Microsoft, призначена для розробників застосунків хмарних обчислень і покликана спростити процес створення онлайнових застосунків.

Історія створення

Корпорація Microsoft представила операційну систему Windows Azure 27 жовтня 2008 року. Представники Microsoft підкреслюють, що Windows Azure – це окремий продукт, а не модифікація якоїсь з існуючих версій Windows.

Windows Azure дозволяє створювати застосунки як за допомогою Microsoft .NET Framework і Visual Studio, так і за допомогою інших інструментів. Операційна система працює на серверах Microsoft, доступ до неї можна отримати за протоколами HTTP, Representational State Transfer (REST), WS-* і Atom Publishing Protocol (AtomPub).

Платформа Azure Services Platform включає п'ять основних компонентів. Це сама операційна система Windows Azure, що керує дисковим простором, застосунками і мережами, і Microsoft SQLServices для роботи з базами даних. Також в платформу входять Microsoft .NET Services, Live Services, і бізнес-компонент, що включає Microsoft SharePoint Services і Microsoft Dynamics CRM Services.

За оцінками аналітиків, анонс Azure є «захисним маневром», щоб нинішні клієнти Microsoft, які користуються її серверами й іншими продуктами, не переходили на хмарні рішення від таких компаній, як Amazon чи IBM, що вже надають низку сервісів для зберігання даних і обчислювальні потужності «в хмарі».

У грудні 2009 Microsoft об'єднав існуючі підрозділи, що займалися Windows Server та Azure, в один Server & Cloud Division (SCD).

Побудова

Платформа виготовлена з групи із трьох технологій, що забезпечують спеціалізований набір можливостей для розробників. Більше того, платформу

Windows Azure можна використовувати в додатках, що працюють локально на комп'ютерах користувачів і додатків, які працюють в хмарі.

Платформа Windows Azure складається з таких компонентів:

– Windows Azure – надає середовище виконання для додатків, заснованих на операційних системах, та на Windows Server, а також місця для зберігання даних. Система працює на віртуальних машинах за допомогою, аналогічної технології Hyper-V.

– Обчислення – відповідає за обчислення розміщення додатків.

– Зберігання – відповідає за зберігання даних в хмарі.

– SQL Azure – надає можливість використовувати реляційну базу даних для запуску в хмарі.

Платформа Windows Azure AppFabric – компонент, який забезпечує додаткову функціональність у вигляді послуг.

Microsoft Azure SQL Database – це хмарний сервіс, що надає можливість зберігання і обробки реляційних даних, а також генерації звітності. На відміну від аналогічних баз даних, бази даних Azure SQL дозволяють користувачам робити реляційні запити до збережених даних, які можуть бути або структурованих або частково структурованих або навіть документів. бази даних SQL має вилучення даних, пошуку, аналізу даних і синхронізацію даних.

Створення додатків

Технології, які можуть бути використані для створення додатків Windows Azure Platform є всі технології, які можуть бути запуснені на платформі Windows. На додаток до технології .net (посилання технології для Windows Azure хмари), ви також можете використовувати такі технології, як: Java, PHP, C/C++ або Python.

Які основні переваги хмарного рішення Windows Azure?

Простота: для створення веб-застосунків, а також управління ними як в хмарних центрах обробки даних, так і в локальних можна застосовувати будь-які звичні мови програмування, такі як: NET, PHP, Java чи Ruby.

Надійність: її забезпечує показник безвідмовності функціонування платформи – 99,9% та можливість цілодобової підтримки від Microsoft.

Безпека: Windows Azure дозволяє для кожного виду бізнесу знайти найбільш ефективне рішення, яке забезпечить необхідний рівень захисту персональних даних чи фінансової інформації.

Ефективність: платформа Windows Azure підвищує продуктивність та ефективність за рахунок скорочення капітальних вкладень. Замовники можуть досягти скорочення операційних витрат на деякі процеси до 30%. При цьому, оплата за обслуговування відбувається на основі фактичного споживання.

Гнучкість: користувачі можуть використовувати переваги засобів розробки, автоматизованого управління службами та глобальної мережі центрів обробки даних, щоб швидше реагувати на потреби клієнтів, концентрувати свої зусилля на зміцненні конкурентних переваг і виходити на нові ринки.

Легкість: користування Windows Azure передбачає внесення змін автоматично і непомітно для користувача, завдяки чому він завжди буде працювати з актуальною версією.

Windows Azure абстрагує ресурси обладнання за допомогою віртуалізації. Кожне додаток, що розгортається в Windows Azure, працює на одній або декількох віртуальних машинах (VM).

Розгорнуті додатки працюють так, ніби вони розташовані на виділеному комп'ютері, хоча можуть використовувати фізичні ресурси, такі як місце на диску, мережевий канал введення-виведення або ядра ЦП, спільно з іншими віртуальними машинами на тому ж фізичному вузлі. Два основних переваги рівня абстракції над фізичним обладнанням – мобільність і масштабованість. Віртуалізація служб дозволяє перенести їх на будь-яку кількість фізичних вузлів в центрі обробки даних.

Поєднання технологій віртуалізації, неспеціалізованого обладнання, багатокористувацьких середовищ і агрегування попиту дозволяють Майкрософт і

нашим клієнтам добиватися значної економії. Це дає можливість оптимізувати завантаження центру обробки даних, тобто збільшувати співвідношення корисної роботи та вартості обладнання і, відповідно, економити значні кошти.

Віртуалізація також дозволяє застосовувати як вертикальне, так і горизонтальне масштабування. Вертикальна масштабованість означає, що в міру зростання потреби ви можете збільшувати кількість ресурсів, наприклад ядер ЦП і пам'яті, на певній віртуальній машині.

Горизонтальна масштабованість означає можливість додавати нові екземпляри віртуальних машин, які є копіями існуючих служб. Для всіх цих примірників застосовується балансування навантаження на мережевому рівні, щоб вхідні запити розподілялися між ними.

На момент написання цього документа платформа Windows Azure включала в себе Windows Azure і SQL Azure.

Windows Azure надає обчислювальне середовище на основі Microsoft® Windows Server® для розміщення додатків, постійне сховище структурованих і неструктурованих даних, а також засоби асинхронного обміну повідомленнями. Крім того, Windows Azure передбачає ряд служб, що дозволяють підключати користувачів і локальні додатки до хмарним додаткам, здійснювати перевірку достовірності, обмін повідомленнями між службами і керування даними, кешування і інші функції, пов'язані з даними.

Windows Azure включає також набір служб управління, які дозволяють контролювати всі ці ресурси за допомогою користувацького веб-інтерфейсу (веб-порталу) або програмними засобами. В більшості випадків є програмний інтерфейс на основі REST, за допомогою якого можна визначити те, як працюватимуть служби. Більшість завдань управління, які можуть бути виконані на веб-порталі, також реалізуються за допомогою API.

По суті справи, SQL Azure являє собою SQL Server®, що працює у вигляді хмарної служби.

І нарешті, існує численний набір засобів і пакетів засобів розробки програмного забезпечення для розробки, тестування та розгортання додатків. Наприклад, розробляти і тестувати додаток ви можете в емульованій локальній

середовищі, створеному за допомогою емулятора обчислень і емулятора сховища. Більшість коштів також інтегровані в такі середовища розробки, як Microsoft Visual Studio®. Крім того, є засоби управління сторонніх постачальників.

Служби і компоненти Windows Azure

Набір служб і компонентів, доступних в Windows Azure і SQL Azure, розрахований на різні конкретні застосування. Після підписки на Windows Azure можна вибрати потрібні можливості, а оплата буде проводитися тільки за використовувані засоби. Додавати компоненти в підписку і видаляти їх з неї можна в будь-який час. Механізм виставлення рахунків для кожної служби залежить від типу наданих можливостей.

Доступні служби і компоненти змінюються в міру розвитку Windows Azure.

Обчислювальне середовище

Обчислювальне середовище Windows Azure складається з платформи для додатків і служб, розміщених в одній або декількох ролях. У Windows Azure можна реалізувати наступні типи ролей.

Емулятор обчислень Windows Azure (веб-ролі і робочі ролі). Додаток Windows Azure складається з однієї або декількох розміщених ролей, запущених в центрах обробки даних Azure.

Як правило, існує хоча б одна веб-роль, доступна користувачам програми. Веб-роль підтримується службами IIS 7.0 і ASP.NET. Додаток може містити додаткові ролі, включаючи робочі, які зазвичай використовуються для виконання фонових задач обробки та підтримки для веб-ролей.

Віртуальна машина (роль VM). Ця роль дозволяє розмістити власний екземпляр операційної системи Windows Server 2008 R2 Enterprise або Windows Server 2008 R2 Standard в центрі обробки даних Windows Azure.

Управління даними

Windows Azure, SQL Azure і пов'язані з ними служби передбачають можливості для зберігання даних і управління ними різними способами. Є наступні служби і компоненти управління даними:

Сховище Windows Azure. Забезпечує чотири основних служби для тривалого зберігання даних в хмарі. Ці служби підтримують інтерфейс REST, доступ до якого здійснюється як з додатків, розміщених в Windows Azure, так і з локальних (віддалених) додатків.

Нижче перераховані чотири служби зберігання:

- служба таблиць Windows Azure надає механізм зберігання даних у вигляді таблиць і підтримує запити для управління даними.
- служба таблиць Azure – це пропозиція NoSQL, що забезпечує зберігання даних без схеми. Вона призначена насамперед для тих випадків, коли необхідно зберігати великі томи даних, при цьому їх потрібно легко витягувати і оновлювати.
- служба великих двійкових об'єктів (BLOB) надає ряд контейнерів, призначених для зберігання текстових і двійкових даних. Передбачені як контейнери блокових великих двійкових об'єктів для потокової передачі даних, так і контейнери сторінкових BLOB-об'єктів для операцій довільного читання і запису.
- служба черг надає механізм для надійного і постійного обміну повідомленнями між екземплярами ролей, наприклад між веб-роллю і робочою роллю.

Диски Windows Azure надають додаткам механізм для підключення одностороннього VHD-диска з файловою системою NTFS як сторінкового великого двійкового об'єкта, а також передачі і завантаження віртуальних дисків через BLOB База даних SQL Azure це хмарна служба Database Services з високим ступенем доступності та масштабованості, заснована на технології SQL Server, яка підтримує популярну модель реляційних баз даних на основі T-SQL. Її можна використовувати з додатками, розміщеними в Windows Azure, і з іншими додатками, що працюють локально або розміщеними на інших майданчиках.

Синхронізація даних. SQL Azure Data Sync – це хмарна служба синхронізації даних, заснована на технологіях Microsoft Sync Framework. Ця служба забезпечує двосторонній синхронізацію даних і надає можливості по управлінню даними, дозволяючи легко розподіляти їх між кількома базами даних SQL Azure, а також між локальними базами даних і базами даних SQL Azure.

Служба кешування. Ця служба надає для додатків розподілений кеш в пам'яті з малою затримкою та високою пропускнуою здатністю, що не вимагає установки і управління, розмір якого динамічно збільшується або зменшується в міру необхідності. Може використовуватися для кешування даних додатків, отримання відомостей про стан сеансів ASP.NET, а також для кешування сторінок ASP.NET

Мережеві служби

Windows Azure містить кілька мережевих служб, які можуть бути використані для оптимізації продуктивності, реалізації перевірки автентичності та поліпшення керованості розміщених додатків.

Це наступні служби.

Мережа кешуючих серверів (CDN). Мережа CDN дозволяє кешувати загальнодоступні статичні дані для додатків в стратегічних точках, розташованих ближче до кінцевим користувачам (з погляду доставки по мережі).

Служба підключення до віртуальної мережі.

Ця служба забезпечує таку настройку ролей додатки, що працює в Windows Azure і на комп'ютерах локальної мережі, що дозволяє уявити, ніби ці ролі розташовані в одній і тій же мережі. При цьому використовується запускається на локальному комп'ютері програмний агент, який встановлює захищене з'єднання IPsec з ролями Windows Azure в хмарі, а також забезпечує можливість адміністрування, відстеження, налагодження та управління ролями безпосередньо.

Диспетчер трафіку віртуальних мереж. Ця служба дозволяє налаштувати перенаправлення запитів і балансування навантаження на основі трьох різних методів.

Як правило, диспетчер трафіку використовується для підвищення продуктивності методом, коли запити перенаправляються до примірника додатки в тому центрі обробки даних, який знаходиться найближче до користувача. Існують і інші методи балансування: відпрацювання відмови і циклічний перебір

Управління доступом (ACS). Ця служба, побудована на основі стандартів, призначена для управління доступом і посвідченнями і використовує безліч постачальників посвідчень (IdP), які можуть виконувати перевірку автентичності користувачів. Служба управління доступом (ACS) діє як служба токенів безпеки

(STS) і допомагає використовувати переваги методів федеративної перевірки автентичності, коли справжність користувача перевіряється не в тій області або домені, де знаходиться додаток. Як приклад можна привести управління доступом користувачів на основі посвідчень, перевіряються постачальником посвідчень, таким як Windows Live® ID або Google

Шина обслуговування. Ця служба забезпечує безпечний обмін повідомленнями і потоками даних в розподілених і гібридних додатках, наприклад для зв'язку розміщених додатків Windows Azure і локальних додатків, виключаючи необхідність реалізації складної інфраструктури брандмауерів і системи безпеки. Вона може використовувати ряд протоколів і шаблонів взаємодії та передачі повідомлень для гарантованої і надійної доставки повідомлень, а також масштабується для різних навантажень і інтегруватися з локальними об'єктами BizTalk Server.

Управління базами даних SQL Azure

Додатки отримують доступ до баз даних SQL Azure так само, як і до локально встановленим екземплярам SQL Server, за допомогою керованих класів доступу до даних ADO.NET, DAAB, OData, власних класів ODBC, PHP, Ruby або технологій доступу до даних JDBC.

Управління базами даних SQL Azure здійснюється через веб-портал, в середовищі SQL Server Management Studio, за допомогою засобів для роботи з базами даних Visual Studio 2010 та інших інструментів для виконання таких завдань, як переміщення і перенесення даних.

Доступні і інші засоби командного рядка для розгортання та адміністрування баз даних. Крім того, доступний диспетчер баз даних, який полегшує роботу з базами даних SQL Azure.

Оновлення додатків для Windows Azure

Після розгортання програми для Windows Azure його необхідно оновлювати при зміні служб ролей у відповідь на нові вимоги та покращення в коді або для виправлення помилок. Ви можете просто виконати повторне розгортання служби. Для цього потрібно призупинити і видалити її, потім виконати розгортання нової версії. Однак ви можете уникнути простою додатків за рахунок поетапного

розгортання (завантаження нового пакету і заміни існуючої робочої версії цим пакетом) або виконання оновлення на місці (відправки нового пакета і його застосування до виконуваних екземплярам служби):

- адміністратори управляють окремими розміщеними службами для підписки Windows Azure. Підписка Windows Azure може включати один або декілька наступних елементів;

- розміщення служби, що складаються з розміщених ролей та примірників в кожній ролі; ролі і екземпляри можуть бути зупинені у робочому режимі або в режимі проміжного зберігання;

- облікові записи зберігання, що складаються з примірників служб таблиць, великих двійкових об'єктів і черг;

- екземпляри CDN.

Що таке CMS

CMS (Content Management System) перекладається як система управління змістом. Основні її функції – зручний інструмент для додавання нової інформації до сайту, редагування та видалення існуючої інформації, управління функціоналом сайту.

Для створення сайту така система не обов'язкова, але для кожного сайту важливе його своєчасне оновлення і розширення. Крім того, повинна бути можливість створення контенту сайту, не маючи навичок програмування.

Тому сьогодні майже усі сайти редагуються за допомогою CMS. Існує багато різних CMS. Їх розрізняють за вільністю використання (безкоштовні, платні та внутрішні), за призначенням (універсальні, призначені для блогів, інтернет-магазинів і т.д.), за способом створення сторінок (попереднє створення, «на льоту» або змішане). Приклади найбільш відомих CMS: Drupal, Joomla!, Wordpress, Magenta.

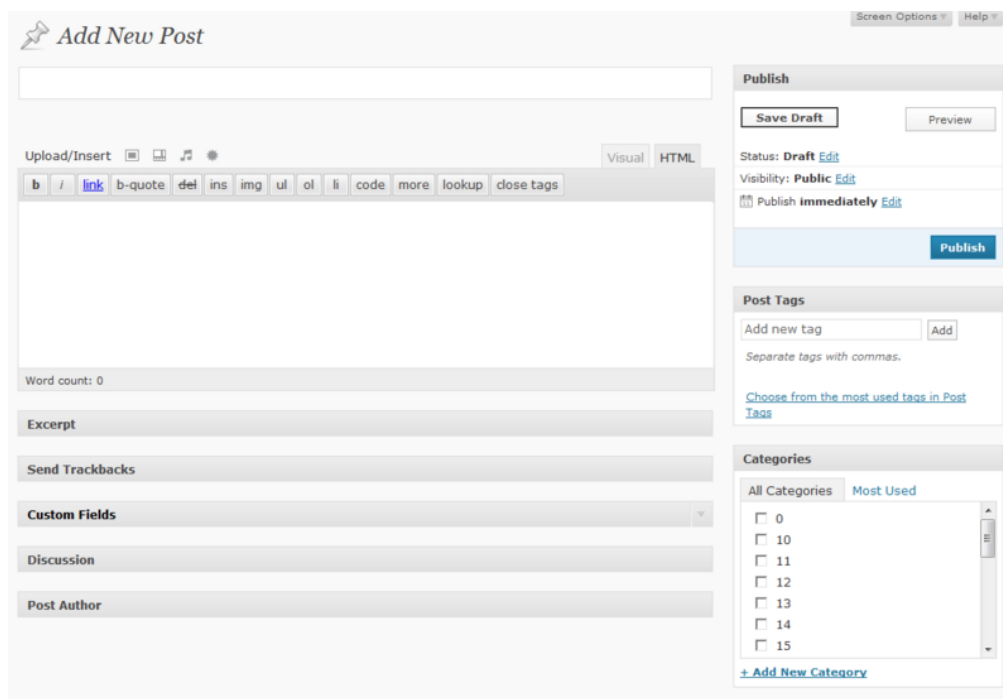
Що таке Wordpress

Wordpress – це CMS, призначена для блогів. Вона вважається однією з найпростіших для користувача, що пояснює її популярність. На основі Wordpress можна створити будь-який сайт, але більшість її функцій призначені саме для

блогів. Однією із переваг є легке налаштування за допомогою численних тем, плагінів та віджетів.

Розглянемо пункти меню зліва.

Posts



Пости редагуються за допомогою редактора, який дозволяє писати у вигляді HTML-коду або у вигляді форматowanego тексту із медіа-елементами. До посту можна додати зображення, відео, звук та flash-елементи.

Кінцевий вигляд посту можна перевірити, натиснувши Preview, і опублікувати на сайті, натиснувши Publish. Посту можна також присвоїти теги, зображення, віднести його до категорії, вказати скорочений зміст.

Media

Тут можна переглянути та змінювати всі зображення, відео, звукові файли, що були завантажені на сервер з метою додання у запис.

Links

Тут переглядаються та редагуються посилання на інші сайти, які можна буде вставляти у відповідних місцях сайту.

Pages

Тут редагуються сторінки сайту. Сторінки схожі на пости, але мають інше призначення: вони відображають статичну інформацію, яка слабо прив'язана до часу. Наприклад, такі розділи як «Про себе», «Контакти» доцільно реалізувати у вигляді сторінок. Сторінки не можуть відноситися до категорії та не мають тегів. Сторінка завжди розглядається окремо, тоді як пости можна вивести у вигляді списку, у тому числі і на сторінці.

Comments

За замовчуванням у відвідувачів сайту є можливість залишати коментарі до записів. Тут їх можна переглядати та редагувати.

Appearance

Управління темами. Теми – це шаблони дизайну сайтів. Саме тема визначає зовнішній вигляд сайту. Є великий вибір тем для Wordpress, безкоштовних та платних. Деякі теми мають можливість налаштування вигляду сайту: зміна шрифтів, гама кольорів, навігації по сайту, а також можливість додання віджетів. Віджети – це невеличкі модулі сайту, що додають нові інтерактивні можливості або візуальні елементи. Прикладами віджетів є хмарка тегів, форма пошуку, календар, міні-чат тощо. Навігація також може бути реалізована за допомогою віджета.

Plugins

Управління плагінами. Плагіни доповнюють функціональність сайту та Wordpress'у, наприклад, фільтр коментарів чи форма відправки повідомлення.

На відміну від віджетів, які розташовуються у певній зоні сторінки сайту, плагіни просто вмикаються і впливають на роботу Wordpress'у, наприклад, заміщуючи деякий текст у сторінці на форму відправки повідомлення. Плагіни, як і теми, також доступні для завантаження.

Users

Управління користувачами. Якщо сайт наповнюють декілька користувачів, доцільно обмежити права на управління сайтом.

Tools

Інструменти, за допомогою яких можна експортувати весь контент на інший сервер з Wordpress або імпортувати контент з інших блогів.

Settings

Різні налаштування сайта, що не залежать від теми.

Обов'язкові файли

Каталог тем Wordpress'у знаходиться у wp-content/. Там треба створити новий каталог з назвою нашої теми. Всередині каталога створити два файли: index.php та style.css. Ці файли мають бути в кожній темі Wordpress. index.php – це шаблон теми за замовченням. Наприклад, якщо переглядається статична сторінка, використовується файл page.php, але якщо такого файла немає, буде використано index.php. Wordpress дозволяє вивести в окремі файли формування різних видів сторінок або навіпаки, розглянути усі варіанти в одному-двох файлах. Файл style.css містить CSS-стилі для усіх сторінок теми та опис теми, наприклад:

Лістинг 2.1 – Файл style.css

```
/*
Theme Name: Twenty Ten
Theme URI: http://wordpress.org/
Description: The 2010 default theme for WordPress.
Author: wordpressdotorg
Author URI: http://wordpress.org/
Version: 1.0
Tags: black, blue, white, two-columns, fixed-width,
custom-header, custom-background, threaded-comments,
sticky-post, translation-ready, microformats,
rtl-language-support, editor-style, custom-menu (optional)

License:
License URI:

General comments (optional).
*/
```

Інші файли шаблонів:

header.php – виводить DOCTYPE, тег head, відкриває теги html і body та виводить верхню частину сайта.

sidebar.php – виводить сайдбар («колонку»), де можуть бути розташовані різні віджети і навігація. Незважаючи на назву, не обов'язково має вигляд колонки.

footer.php – виводить нижню частину сайта, закриває теги body і html.

functions.php – об'явлення нових функцій, що використовуються в темі, а також деяких елементів теми: навігації, сайдбарів.

page.php – виводить сторінку.

single.php – виводить окремий пост.

Є ще багато інших шаблонів, також є можливість створювати свої. Для данної теми достатньо файлів header.php, footer.php, functions.php.

Вміст index.php

```
<?php
    get_header();
?>
<div id="content">
    <div id="content_top" <?php if (is_front_page())
        echo 'class="home"' ?>>
        <div id="content_top_center" <?php if (is_front_page())
            echo 'class="home"' ?>>
            <?php
                if (!is_front_page()) {
                    echo '<div id="page_title">';
                    the_title();
                    echo '</div>';
                } else {
                    $top_page = get_page_by_title('Home (top)');
                    echo $top_page->post_content;
                }
            ?>
        </div>
    </div>
    <div id="content_bottom" <?php if (is_front_page())
        echo 'class="home"' ?>>
        <div id="content_bottom_center" >
            <?php
                if (have_posts()) : while (have_posts()) : the_post();
                    the_content();
                endwhile;
            endif;
        ?>
    </div>
</div>
<?php
    get_footer();
?>
```

У цьому файлі виводиться лише центральна частина сторінки, верхня і нижня частини виводяться з файлів header.php і footer.php за допомогою функцій get_header(), get_footer(). Зверніть увагу на таку конструкцію:

```
if (have_posts()) : while (have_posts()) : the_post();
    the_content();
endwhile;
endif;
```

Це так званий Цикл Wordpress ("The Loop"). Він виводить усі пости (у даному випадку лише зміст сторінки). Є спеціальні функції, що використовуються тільки всередині Циклу і дозволяють керувати виводом постів на сторінку.

Вміст header.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta content="charset=<?php bloginfo('charset'); ?>" >
    <title><?php bloginfo('name'); ?></title>
    <link rel="stylesheet" type="text/css" media="all" href="<?php bloginfo('stylesheet_url'); ?>" >
    <?php wp_head(); ?>
  </head>
  <body>
    <div id="header">
      <div id="header_center">
        <div id="header_center_img">
          <div id="menu_sidebar">
            <?php
              dynamic_sidebar('menu-sidebar');
            ?>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

Кодування та заголовки беруться з бази даних за допомогою функції `bloginfo`. Зверніть увагу на функцію `wp_head` – через неї плагіни можуть вставляти свої скрипти та стилі, тому вона потрібна у будь-якій темі. Функцією `dynamic_sidebar` у верхню частину сайту вставляється сайдбар, в якому буде навігаційне меню.

Вміст footer.php

```
<div id="footer">
  <div id="footer_center">
    <?php
      dynamic_sidebar('footer-sidebar');
    ?>
    <a href="#"></a>
  </div>
</div>
<?php
  wp_footer();
?>
</body>
```

З такою ж метою тут викликається функція `wp_head`. Копірайт та контактна інформація виводяться через сайдбар.

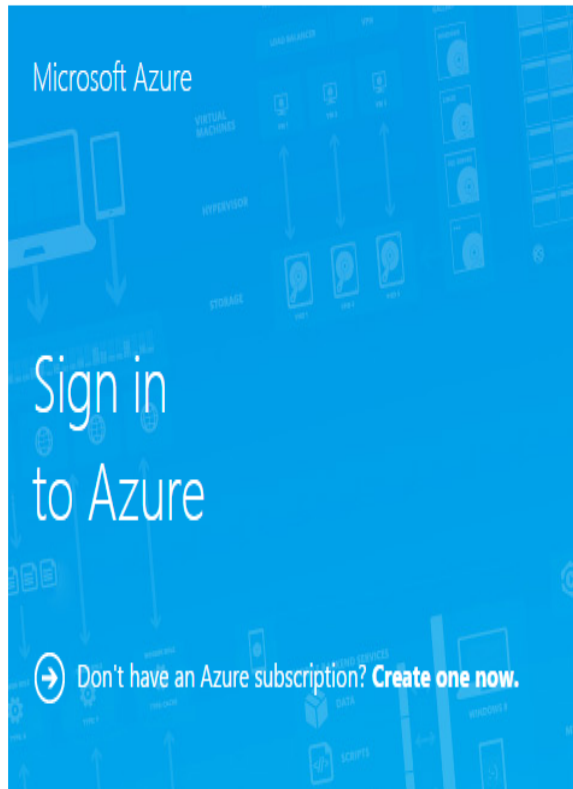
```
<div id="footer">
  <div id="footer_center">
    <?php
      dynamic_sidebar('footer-sidebar');
    ?>
    <a href="#"></a>
  </div>
</div>
<?php
  wp_footer();
?>
</body>
</html>
```

Тут створюються навігаційне меню і обидва сайдбари.

ПОРЯДОК ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Виконайте вхід на порталі Azure.

oration [US] <https://login.live.com/login.srf?wa=wsignin1.0&wtrealm=urn%3afederation%3aMicrosoftOnline&wctx=estsredirect%3d2>



Вхід

Обліковий запис Microsoft [Що це?](#)

smartboy2611@gmail.com

Пароль

☐ Не виходити

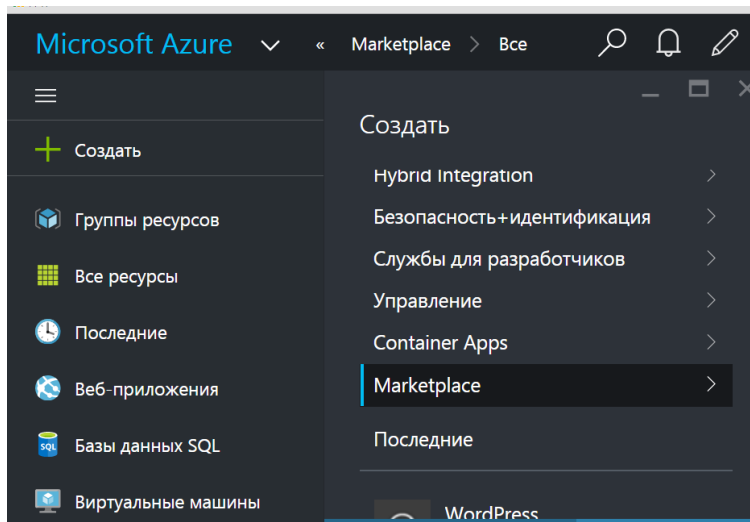
Увійти

[Не можете отримати доступ до свого облікового запису?](#)

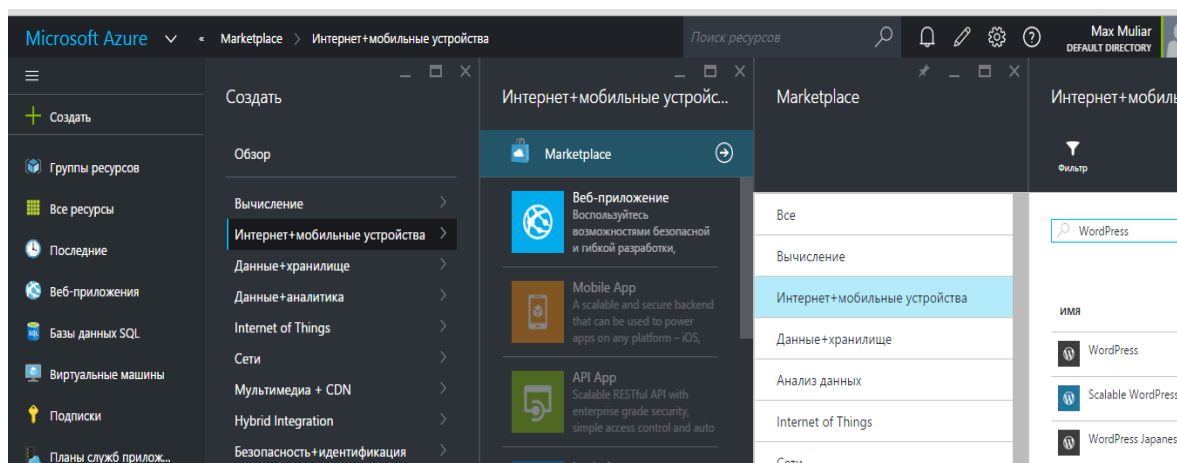
2. Відкрийте Azure Marketplace, клацнувши значок Marketplace.



Крім того, можна клацнути значок Створити в правому верхньому куті приладової панелі і вибрати Marketplace в нижній частині списку.

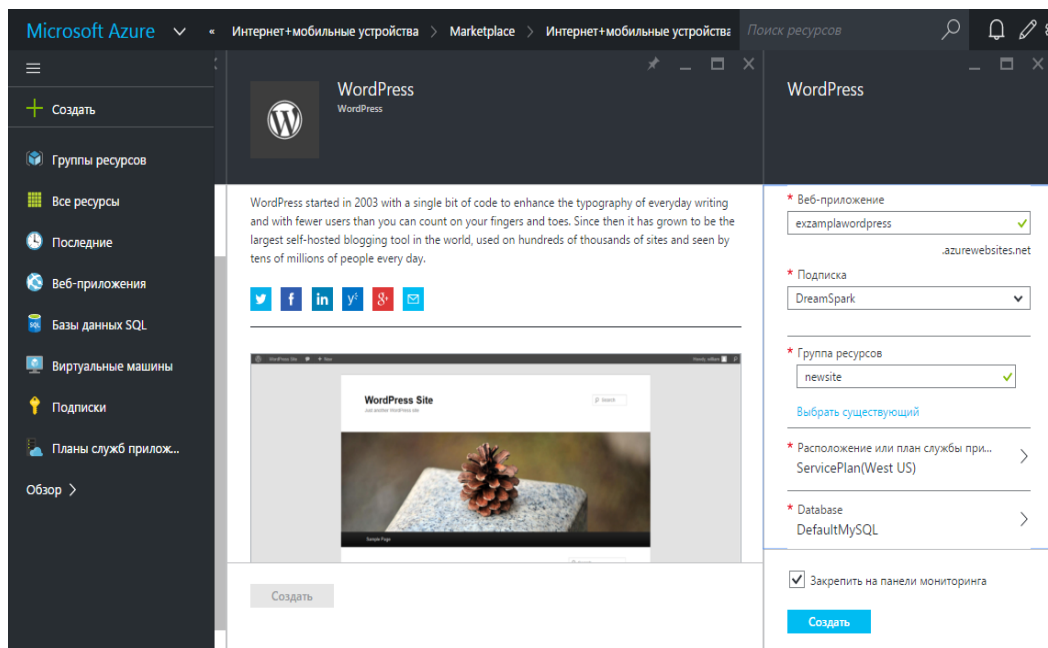


3. Виберіть Інтернет + мобільні пристрої. Знайдіть WordPress і натисніть значок WordPress



4. Прочитавши опис додатки WordPress, натисніть Створити.

5. Клацніть Веб-додатки і вкажіть необхідні значення для налаштування веб-додатки.

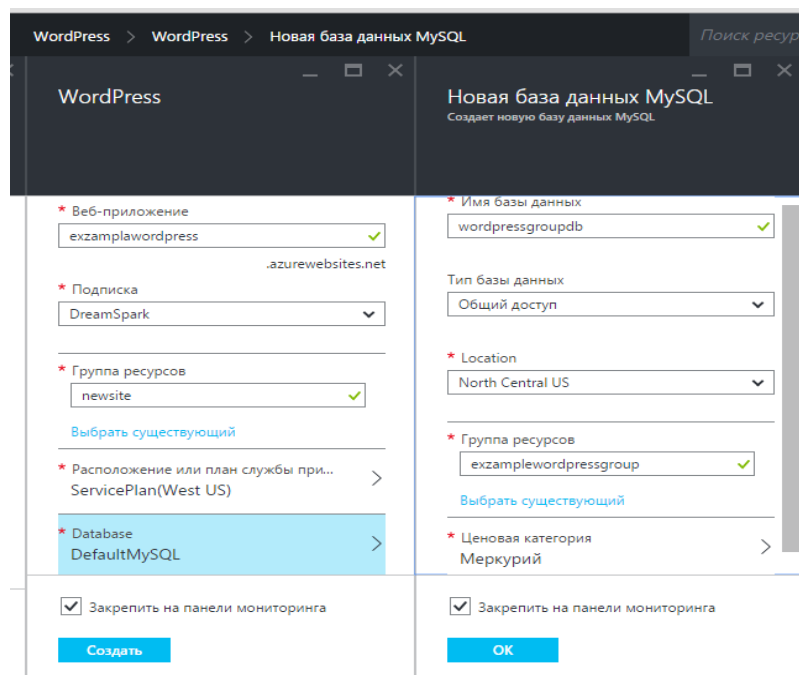


6. Нажміть на елемент База даних і вкажіть в колонці Нова база даних MySQL необхідні значення для налаштування бази даних MySQL.

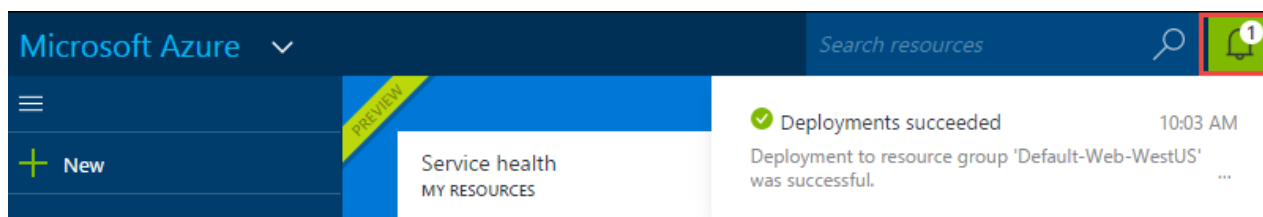
- Введіть нове ім'я або використайте стандартне ім'я;
- поле Тип бази даних залишіть стандартне значення;
- виберіть цінову категорію. Для цього нам підходить «Меркурій» (безкоштовний з мінімум дозволених підключень і місця на диску).

7. У колонці **Нова база даних MySQL** натисніть ОК.

8. У колонці **WordPress** прийняти умови й натисніть кнопку Створити.



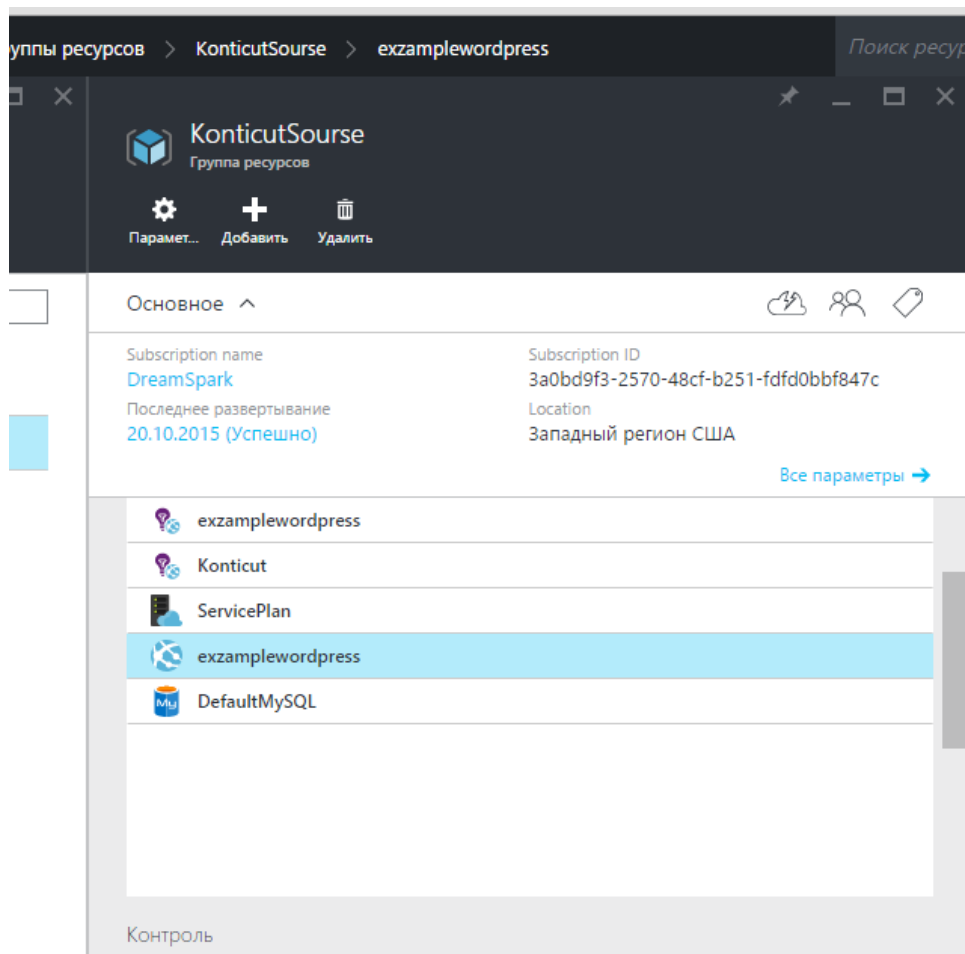
Служба додатків Azure створить веб-додаток (зазвичай це займає менше хвилини). Можна відслідковувати хід виконання, клацнувши значок дзвіночка у верхній частині сторінки порталу.



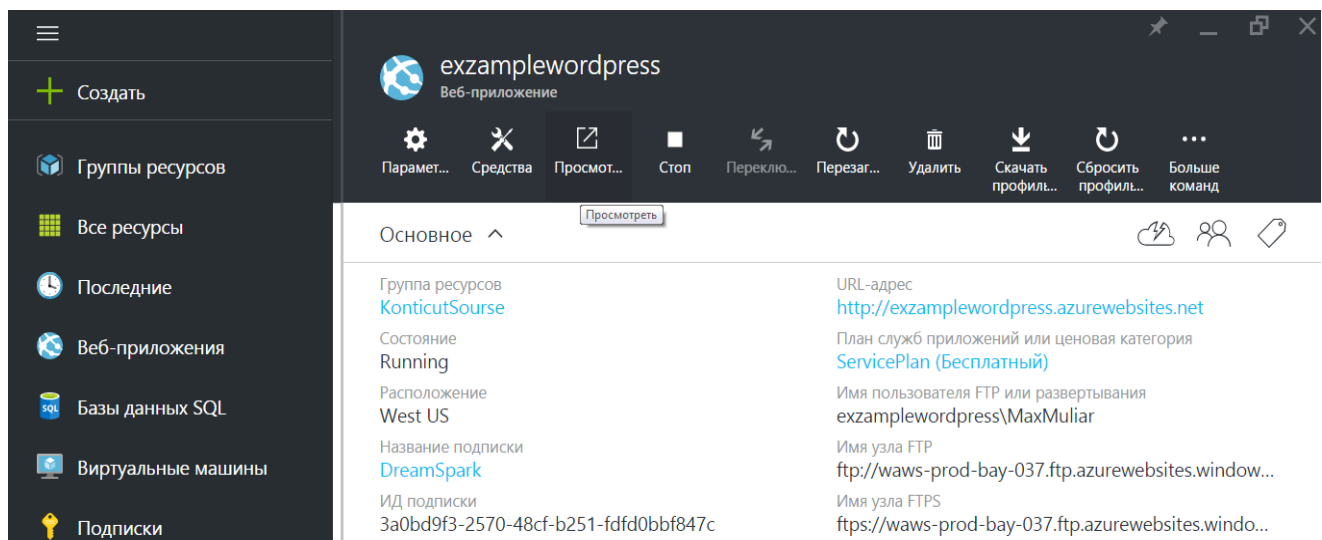
Запуск веб-додатка WordPress і управління ним

1. Після завершення створеного веб-додатку перейдіть на порталі до групи ресурсів, в якій було створено програму, і ви побачите веб-додаток і базу даних.

- У колонці **Група ресурсів** нажміть на рядок веб-додатки.
- Нажміть на веб-додаток для перегляду відомостей про програму.



– В колонці веб-додатку натисніть кнопку **Огляд**

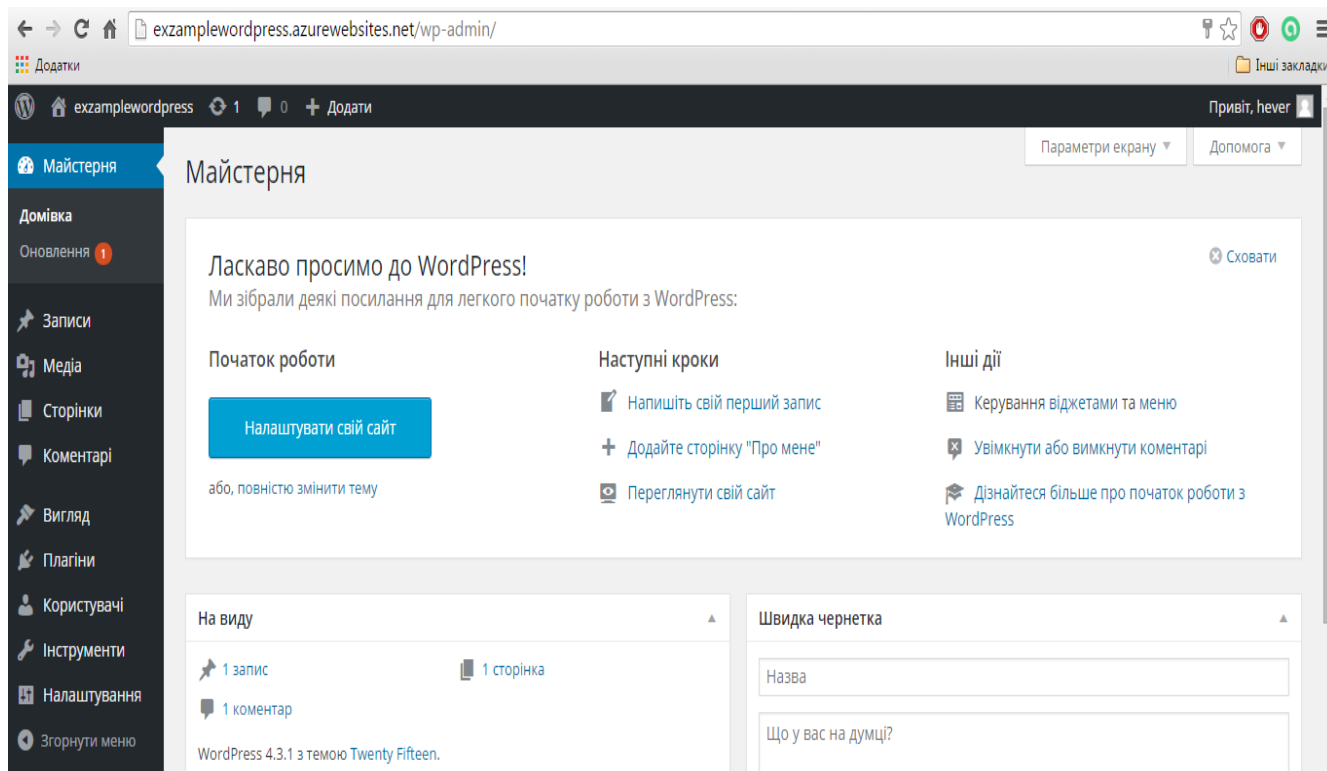


2 На сторінці **вітання** WordPress введіть відомості про конфігурацію, які потрібні WordPress, після чого натисніть кнопку **Встановити WordPress**.

The screenshot shows the WordPress 'Welcome' screen during installation. At the top is the WordPress logo and the word 'WordPress'. Below it is a 'Welcome' heading followed by a paragraph: 'Welcome to the famous five minute WordPress installation process! You may want to browse the [ReadMe documentation](#) at your leisure. Otherwise, just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.' The next section is 'Information needed', with a note: 'Please provide the following information. Don't worry, you can always change these settings later.' Below this are several form fields: 'Site Title' (empty), 'Username' (filled with 'admin', with a note: 'Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol'), 'Password, twice' (two empty fields, with a note: 'A password will be automatically generated for you if you leave this blank.' and a 'Strength indicator' button), 'Your E-mail' (empty, with a note: 'Double-check your email address before continuing.'), and 'Privacy' (a checkbox labeled 'Allow search engines to index this site.' which is checked). At the bottom is an 'Install WordPress' button.

3 Увійдіть, використовуючи облікові дані, створені на сторінці вітання.

4 Відкриється сторінка панелі моніторингу вашого сайту.

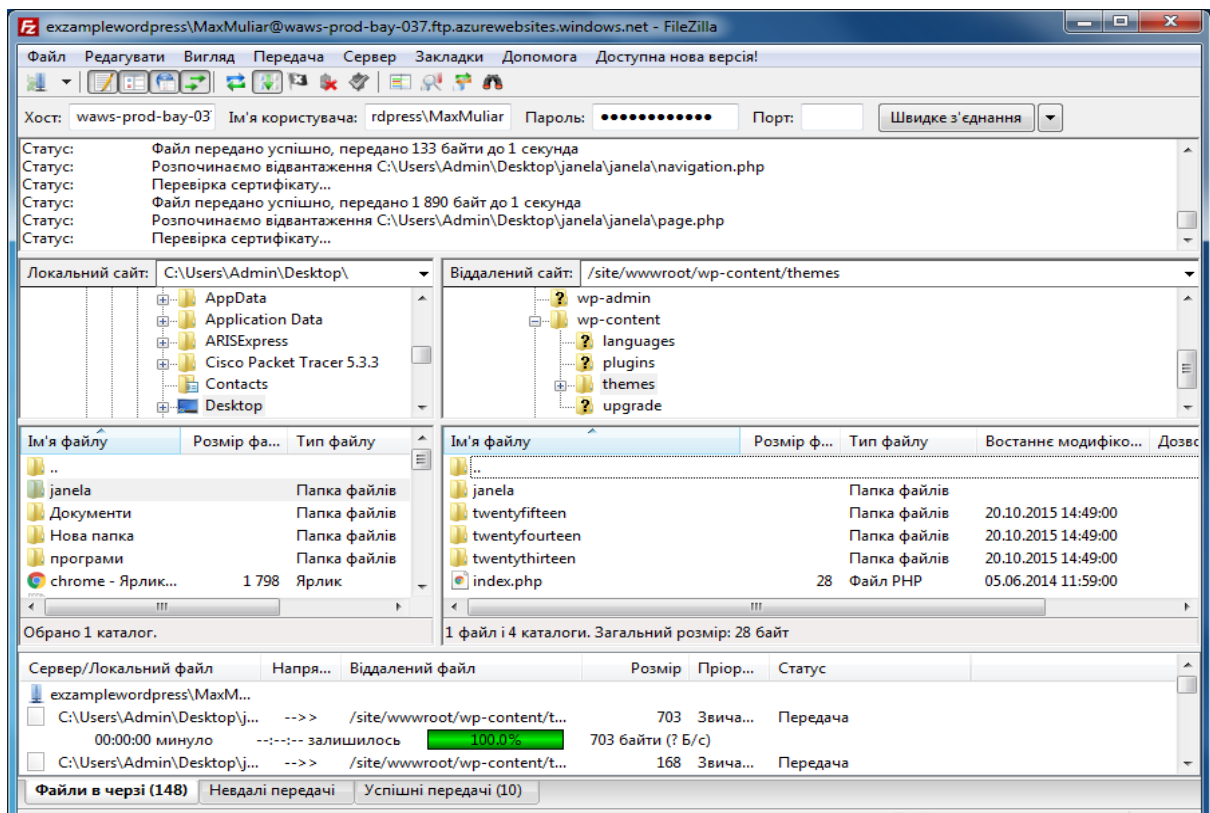


5 Приклад перенесення веб-сайту на хостинг.

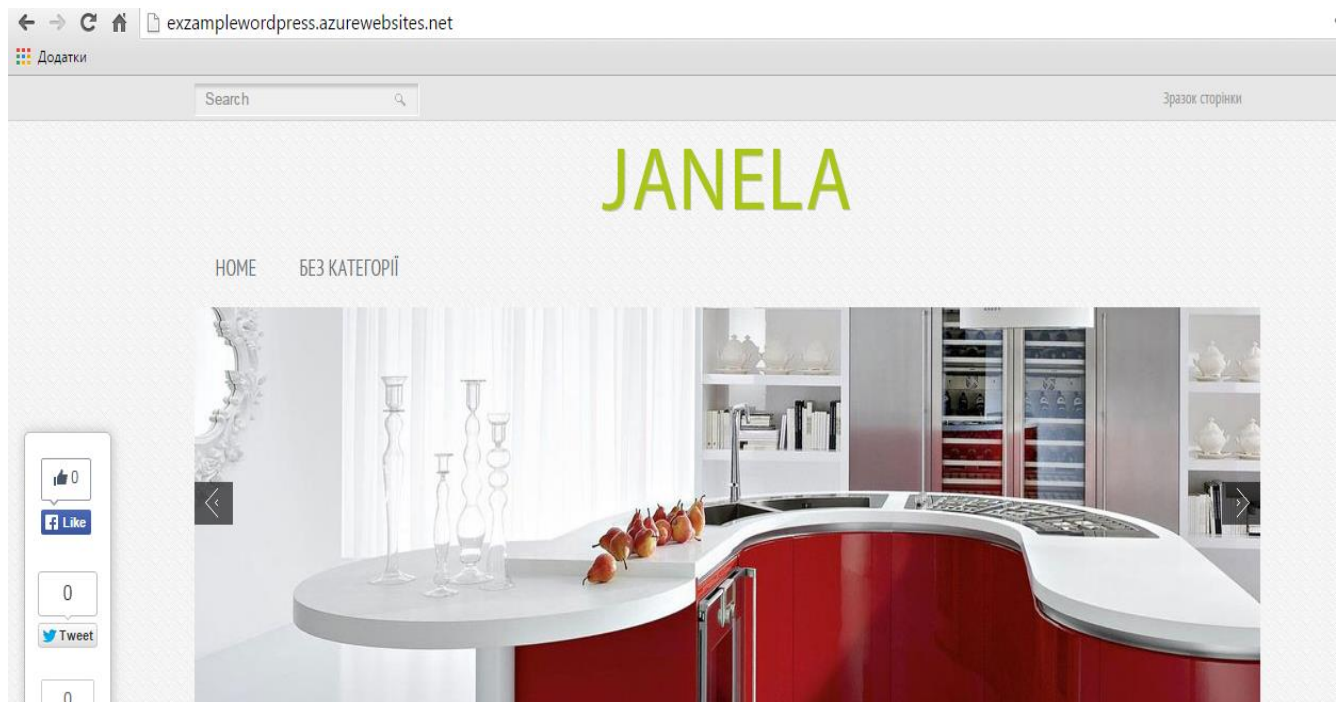
За допомогою попередньо встановленої програми FileZilla було встановлене з'єднання з FTP та завантажений сайт. В якості прикладу хосту та ім'я користувача будемо використовувати:



Приклад приєднання до FTP веб-сайту за допомогою FileZilla зображений на рисунку нижче.



Сайт запущений та функціонує, але все одно потрібно буде перевірити чи всі його елементи, функціонують правильно.



Запитання до лабораторної

1. Microsoft Azure SQL Database це – ?
2. Що означає вертикальна масштабованість?
3. Що означає горизонтальна масштабованість?
4. Чи підтримується веб-роль службами IIS 7.0 і ASP.NET?
5. Яку кількість основних служб для тривалого зберігання даних в хмарі забезпечує Сховище Windows Azure?
6. Чи передбачають служби можливості для зберігання даних і управління ними різними способами ? (обґрунтуйте).

Список використаних джерел

1. https://uk.wikipedia.org/wiki/Windows_Azure.
2. [https://msdn.microsoft.com/ru-ru/library/hh680911\(v=pandp.50\).aspx](https://msdn.microsoft.com/ru-ru/library/hh680911(v=pandp.50).aspx).
3. <https://uk.wikipedia.org/wiki/WordPress>.

ЛАБОРАТОРНА РОБОТА № 3

ТЕМА: РОЗГОРТАННЯ ВЕБ-САЙТУ НА AZURE ЗА ДОПОМОГОЮ MICROSOFT VISUAL STUDIO

Порівняльний аналіз інфраструктури сучасних хмарних-систем

Архітектуру хмарного обчислювального середовища можна розділити на чотири рівні: рівень апаратних засобів дата-центру, рівень інфраструктури, рівень платформи і прикладний рівень, як показано на рис.1.3.

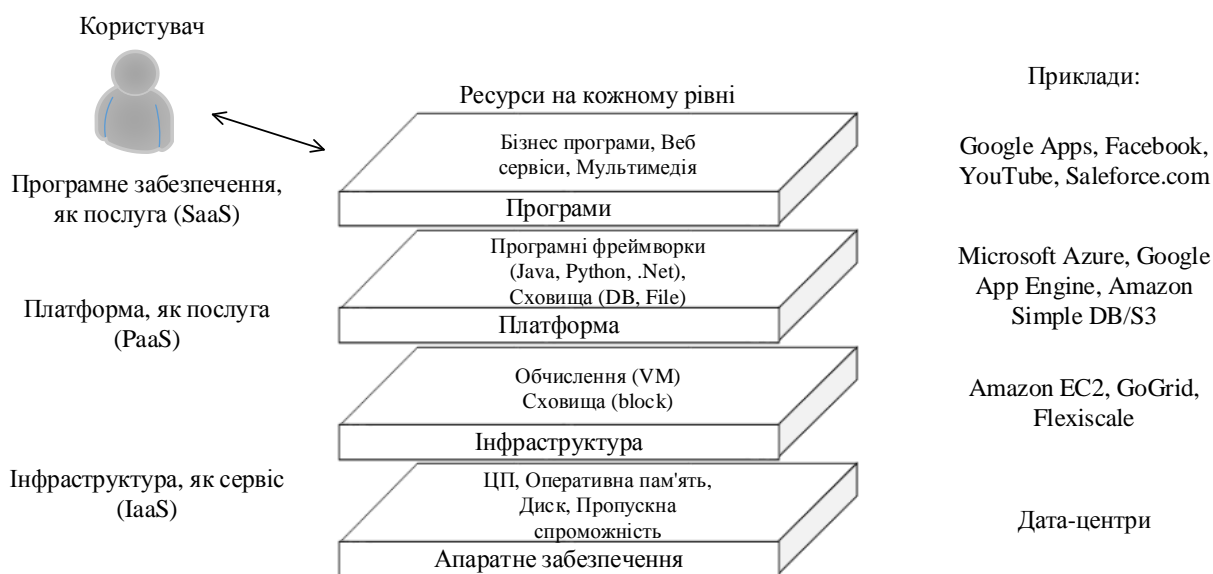


Рисунок 1.3. Багаторівнева модель хмарних обчислень

Опишемо кожний з рівнів детально.

Апаратний рівень: цей рівень відповідає за управління фізичними ресурсами хмари, у тому числі фізичних серверів, маршрутизаторів, комутаторів, систем електроживлення та охолодження. На практиці, апаратний рівень зазвичай реалізується в центрах обробки даних. Центри обробки даних звичайно містять тисячі серверів, які організовані в стійках та з'єднані через свічі, маршрутизатори або інші пристрої. Типові проблеми на апаратному рівні включають апаратну конфігурацію, відмовостійкість, управління трафіком, управління електроживленням та охолодженням ресурсів.

Рівень інфраструктури: також відомий як рівень віртуалізації. Цей рівень створює пул зберігальних і обчислювальних ресурсів шляхом поділу фізичних ресурсів з використанням технологій віртуалізації, таких як Xen, KVM і VMware. Рівень інфраструктури – важливий компонент хмарних обчислень, так як багато ключових функцій, таких як динамічний розподіл ресурсів, надаються лише за допомогою технологій віртуалізації.

Рівень платформи: створений поверх рівня інфраструктури, цей рівень складається з операційних систем і програмних фреймворків. Метою цього рівня є зведення до мінімуму зусиль з розгортання програм безпосередньо в контейнери віртуальних машин. Наприклад, Google App Engine працює на рівні платформи і підтримує API реалізації сховища, бази даних і бізнес-логіки типової веб-програми.

Прикладний рівень: на найвищому рівні ієрархії, цей рівень складається з фактичних програм у хмарі. На відміну від традиційних програм, хмарні програми можуть використовувати автоматичні функції масштабування для досягнення більш високої продуктивності, доступності і низької вартості експлуатації.

У порівнянні з традиційними хостинг середовищами, такими як виділені серверні ферми, архітектура хмарних обчислень є більш модульною. Кожний рівень має слабкий зв'язок з рівнями вище і нижче, дозволяючи кожному рівню розвиватися окремо. Архітектурна модульність дозволяє хмарним обчисленням мати широкий діапазон використання при одночасному зниженні вимог на управління та витрат на обслуговування.

Розглянемо інфраструктури сучасних Cloud-систем більш детально на прикладі трьох самих популярних рішень: Amazon EC2, Google App Engine та Microsoft Windows Azure platform.

Amazon EC2. Amazon Web Services (AWS) являє собою набір хмарних сервісів, надання хмарних обчислень, зберігання та інші функції, які дозволяють організаціям і приватним особам розгортання програм і послуг на довгостроковій основі і на вимогу за фіксованими цінами. Amazon Web Services доступні через HTTP, з використанням REST і SOAP протоколів. Amazon Elastic Compute Cloud (Amazon EC2) дозволяє користувачам запускати і керувати зразками серверів в дата центрах хмари, використовуючи API або доступні інструменти і утиліт. Зразки

віртуальних машин EC2 працюють на системі віртуалізації Xen. Після створення і запуску екземпляра, користувачі можуть завантажувати програмне забезпечення і вносити до нього зміни. Користувачі мають майже повний контроль над всім стеком програмного забезпечення. EC2 надає можливість розміщувати примірники у декількох місцях. Локації EC2 складаються з регіонів і зон. Регіони складатися з однієї або більше зон, відокремлені один від одного географічно. Зони розташовані в різних місцях, які розроблені, так щоб бути ізольованими від відмов в інших зонах і забезпечити підключення до мережі в іншій зоні у тому ж регіоні з низькою затримкою. Образи EC2 машин зберігаються в Amazon Simple Storage Service (Amazon S3). S3 зберігає дані як "об'єкти", які згруповані в "бакети". Кожний об'єкт містить від 1 байта до 5 гігабайт даних. Імена об'єктів по суті являються URI лінками. Amazon Virtual Private Cloud (VPC) є безпечним каналом між існуючою IT-інфраструктурою компанії та хмарою AWS. Amazon VPC дозволяє підприємствам підключити існуючу інфраструктуру до багатьох ізольованих AWS обчислювальних ресурсів за допомогою віртуальної приватної мережі (VPN) зв'язку, а також розширити свої існуючі можливості з управління, такі як службою безпеки, брандмауером і системою виявлення вторгнень, щоб включати їх AWS ресурсів. Метрики про EC2 включають, наприклад, завантаження процесора, мережі, швидкість вхід/вихід байт, диск операції читання/запису, і т.д.

Google App Engine. Google App Engine представляє собою платформу для традиційних веб-програм в керованих центрах обробки даних Google. В даний час допустимими мови програмування в Google App Engine: Python, Java і Go. Веб-платформи, які працюють на платформі Google App Engine включають Django, CherryPy, Pylons, і web2py, а також власний фреймворк, написаний Google для веб-програм та схожий на JSP або ASP.NET. Google виконує розгортання коду в кластер, моніторинг, відновлення після збою і запуск екземпляра програми, по необхідності. Поточна API підтримує такі функції, як зберігання і вилучення даних з BigTable (нереляційна бази даних), відправка HTTP запитів і робота з кешем. Розробники мають доступ до файлової системи на App Engine тільки на читання. Платформа Google App Engine дозволяє користувачам запускати і розміщувати свої веб-програми в інфраструктурі Google. Такі програми легко будувати, підтримувати і

масштабувати, в той час коли потрібен трафік чи місце для зберігання даних. При використанні Google App Engine, немає необхідності в підтриманні серверів і в команді адміністраторів. Google також надають можливість користувачеві по обмеженню доступу до програми для членів організації або до всього світу. За допомогою цієї технології створюються програми, які надійно працюють під великим навантаженням і використовують великі обсяги даних.

Microsoft Windows Azure platform. Microsoft Windows Azure Platform складається з трьох компонентів, кожний з яких забезпечує конкретний набір послуг для користувачів хмари. Windows Azure надає Windows-based середовище для запуску програм і зберігання даних на серверах в дата центрах Microsoft. SQL Azure надає послуги передачі даних в хмарі на основі сервера SQL Server. NET Services пропонує розподілену інфраструктуру послуг хмарних програм та програм, які виконуються на локальних машинах. Windows Azure платформа може бути використана як для програм, що працюють в хмарі так і для програм, які виконуються на локальних машинах. Windows Azure підтримує програми, побудовані на .NET Framework та інших мовах програмування, які підтримуються операційною системою Windows (наприклад, C#, Visual Basic, C++ та інші). Windows Azure дозволяє зберігати дані в блоках, таблицях і чергах, використовуючи RESTful API через HTTP або HTTPS. Всі фізичні ресурси, такі як віртуальні машини і програми в дата центрі контролюються програмним забезпеченням під назвою «Fabric контролер». З кожним додатком, користувачі завантажують файл конфігурації, який забезпечує на основі XML опис того, що програма вимагає для роботи. На основі цього файлу, Fabric контролер вирішує, де нові програми повинні працювати, вибираючи з фізичних серверів для оптимізації використання обладнання. Клієнти хмарних сховищ часто виявляються зав'язаними на конкретного провайдера хмарних сховищ, і це має певні ризики. Це призводить до свого роду "інерції даних" – чим більше даних зберігається у одного провайдера, тим важче стає його змінити. Клієнти не будуть в змозі вільно переміщатися до нових хмарних сховищ: це може завести в пастку “залишених даних”.

Таблиця 3.1.

Порівняння можливостей основних хмарних систем

| Хмарний провайдер | Amazon | Google App Engine | Windows Azure |
|--|---|---|--|
| Модель обслуговування | Iaas | PaaS | PaaS, IaaS |
| Тип програм | Прикладні програми | Традиційні веб-програми | Прикладні Windows програми |
| Обчислення | Рівень ОС на ВМ з Xen гіпервізором | Існуючий веб фреймворк | Середина запуску Microsoft CLR |
| Сховище | Еластичне сховище блоків; Amazon Simple Storage Services (S3); SimpleDB | BigTable та MegaStore | Azure сервіси сховищ та SQL сервіси даних |
| Автоматичне масштабування | Автоматичне змінення кількості запусканих екземплярів | Автоматичне масштабування, яке є прозорим для користувача | Автоматичне масштабування на основі ролей програми та конфігураційного файлу |
| Підтримка стандартного інтерфейсу | Немає | Немає | Немає |
| Організація безпечного доступу | Базовий рівень, через списки доступу | Аудит системи, списки доступу | Списки доступу, аутентифікація користувачів і процесів, обмеження права на запис |
| Міграція до іншого провайдера | Платна | Платна | Платна |
| Використання сторонніх засобів для доступу до ресурсів | Так | Ні | Так |

Хоча хмарні провайдери дотримуватися суворих угод про рівень обслуговування (SLA) з вражаючим часом безвідмовної роботи і часу відгуку, невдачі і простої все ж трапляються. Крім того є випадки збоїв у провайдерів

хмарних послуг можуть призвести до втрати даних маси клієнтів, і хоча такі випадки доволі рідкість, вони можуть тривати до декількох годин.

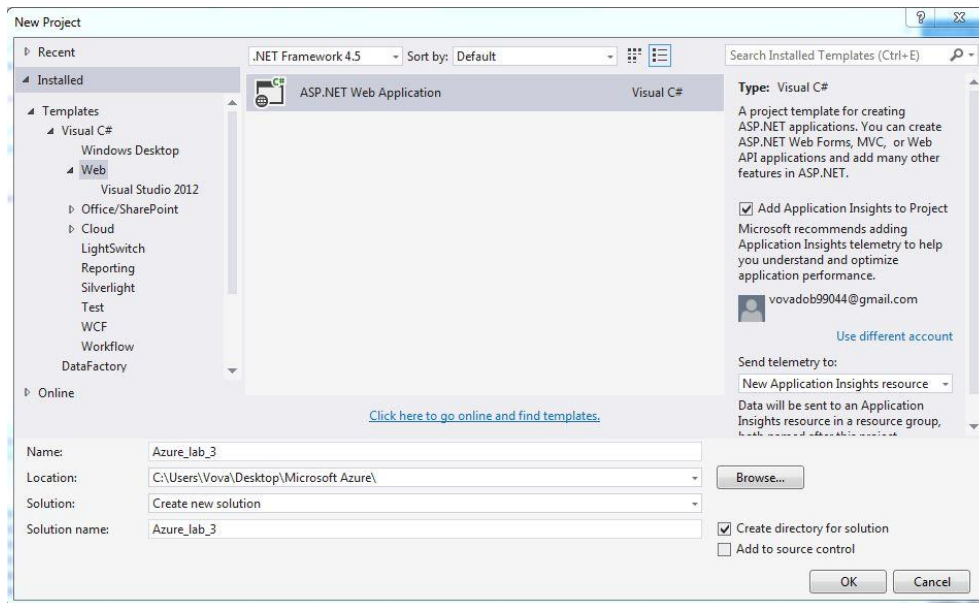
У табл. 3.1 наведено порівняння характеристик популярних хмарних сервісів (Amazon Web Services, Google app engine та Windows Azure Cloud) в термінах класів обчислень, типів цілей застосування, і їх моделей обчислень, зберігання та автоматичного масштабування. Ці хмарні системи засновані на різних рівнях абстракції та управління ресурсами. Користувачі можуть вибрати один тип або комбінацію з декількох типів хмарних сервісів, щоб задовольнити специфічним бізнес-вимогам.

Також можна побачити ряд проблем, з якими стикаються користувачі хмарних систем. Хмарні провайдери не мають ніякого уніфікованого інтерфейсу, який можна було б використати для легкого переходу між різними хмарними провайдерами, або для забезпечення додаткової відмовостійкості. Різні хмарні провайдери надають різні функції і рівні для контролю безпеки для своїх користувачів, але здебільшого такий функціонал є обмеженим на ринку хмарних систем.

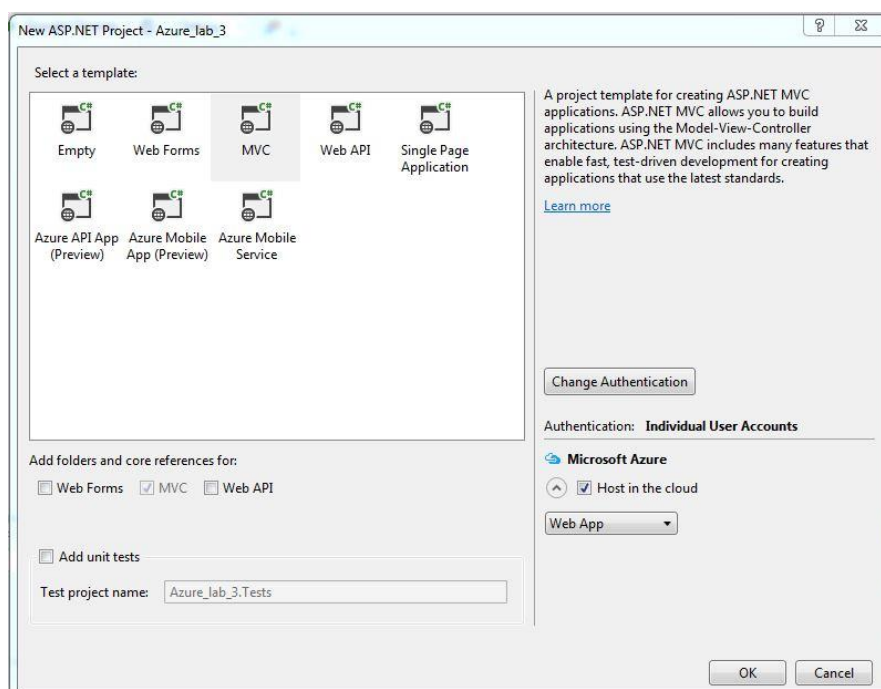
Існує декілька рішень вирішення існуючих проблем хмарних систем. Наприклад компанія Intel пропонує використовувати метод анонімізації даних для підвищення рівня їх безпеки при зберіганні в публічних хмарних системах. Цей підхід передбачає підстановку певних даних замість конфіденційних та створення таблиці відповідності. У цього підходу є недоліки: використання його потребує додаткових затрат на реалізацію підходу для структури даних користувача, при використанні такого підходу досить легко дізнатись початкові дані, наприклад, додавши свої дані у систему і подивившись на використаний ідентифікатор в сховищі.

Порядок виконання лабораторної роботи

1. Лабораторна робота була розроблена в Visual Studio 2013, для розгортання сайту на Azure, потрібно встановити оновлення VS2013.5 з сайту <https://www.microsoft.com/en-us/download/confirmation.aspx?id=48129>.
2. Далі створюємо проект у Visual Studio, веб-сайт на мові програмування C# використовуючи технологію створення веб-сайтів і веб-сервісів ASP.NET.



3. Вибираємо шаблон створення веб-сайту за замовчуванням використовується MVC.

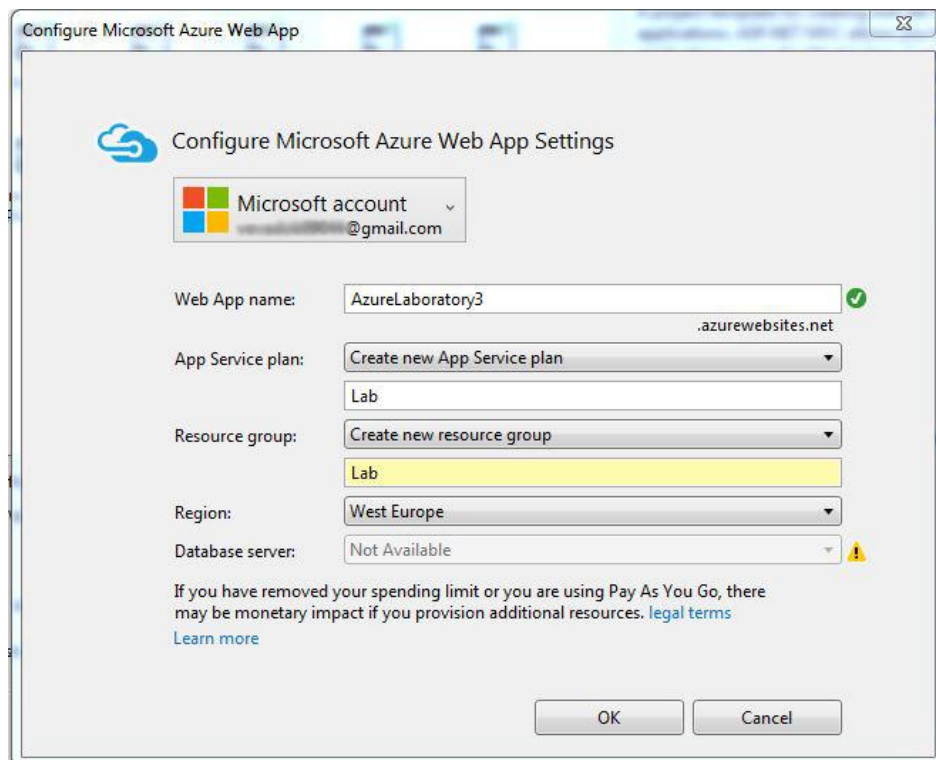


Для перевірки чи встановлено у вашій VS оновлення для розміщення сайту на хостінгеру Microsoft Azure, повинно відображатися вікно.



У даному випадку розгортаємо сайт як **Web app**, але можливі і інші можливості розгортання на **Virtual Machine**, але наш аккаунт на MS Azure не надає можливості для розгортання на **Virtual Machine**. Тому обираємо **Web app**.

4. Налаштування вікна для конфігурації MS Azure Web App.

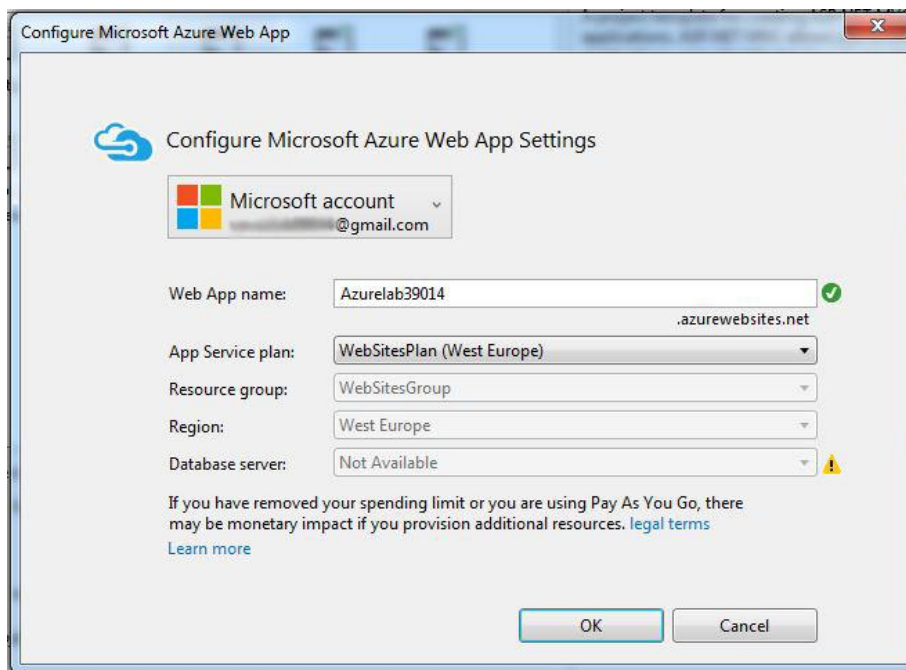


В цьому вікні відображається:

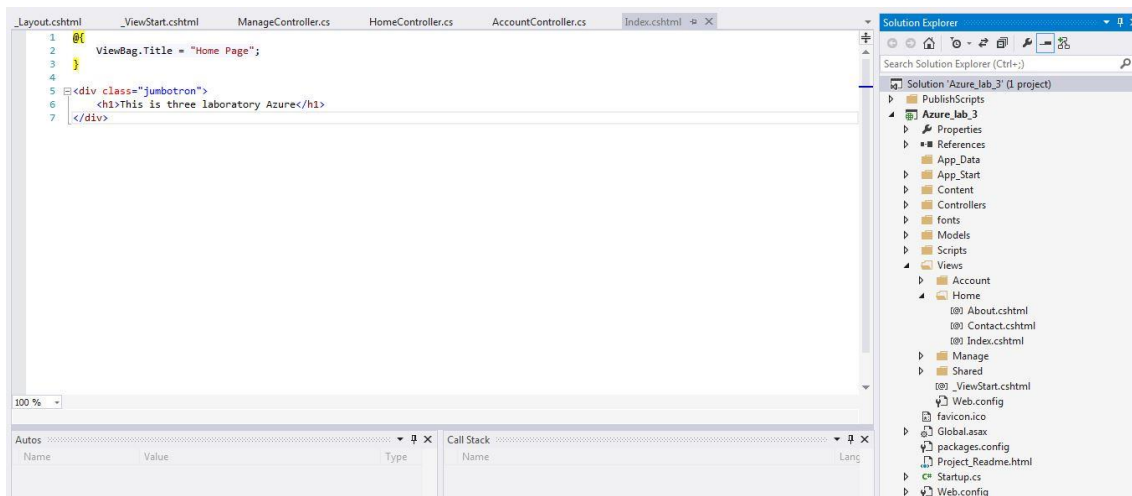
- Ім'я аккаунта користувача (**MS account**);
- Ім'я сайту за, яким він буде розміщуватися на *hostinger Azure* (**Web App name**);
- Ім'я сервіс-плана, яке можна встановити самому(**App Service plan**);
- Вказати назву групі ресурсів(**Resource group**);

- Вибрати із списку регіон в якому розташовані(**Region**);
- Вказати базу даних(якщо використовується в проєкті) (**Database service**).

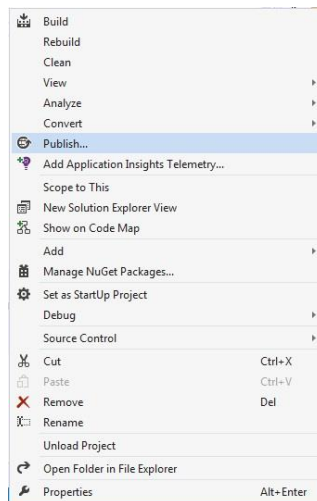
Якщо ви не в перший раз розгортаєте свої проєкти на Azure, потрібно заповнити поле **Web App** і за шаблоном вибрати **Service plan**.



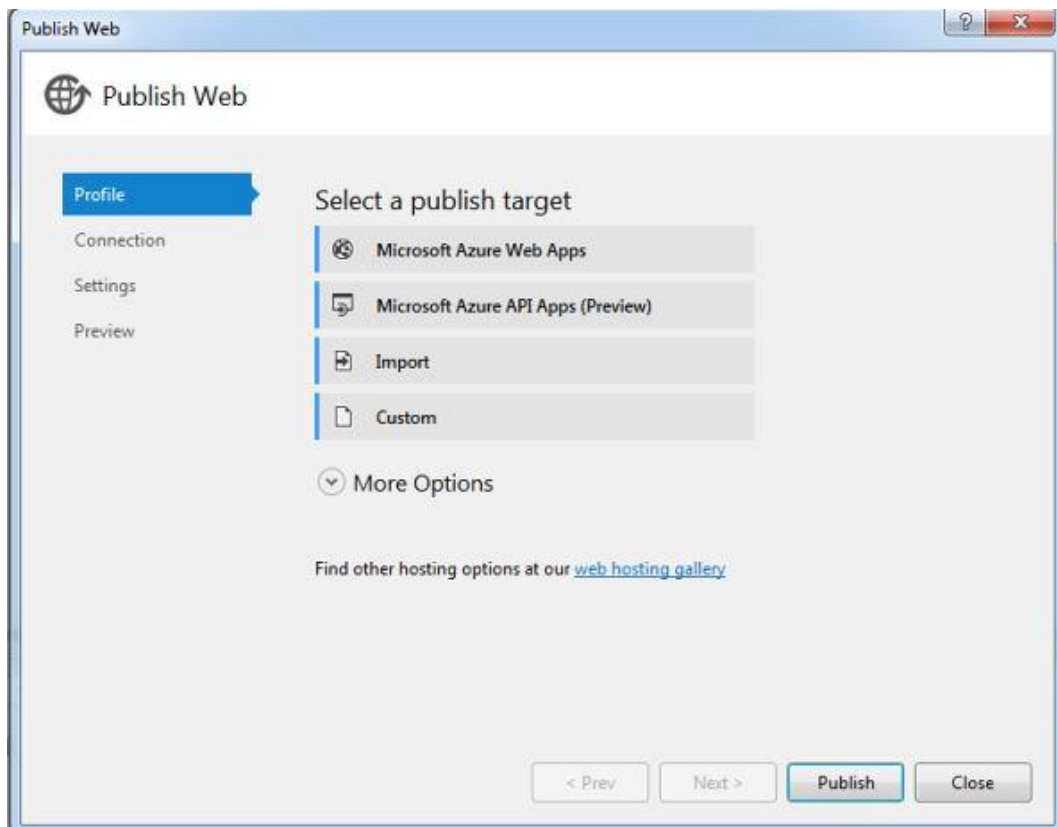
Після конфігурацій Azure Web App, в проєкт вносимо зміни в код, або пишемо з нуля сайт. В залежності від вибраного шаблона створеного сайту.



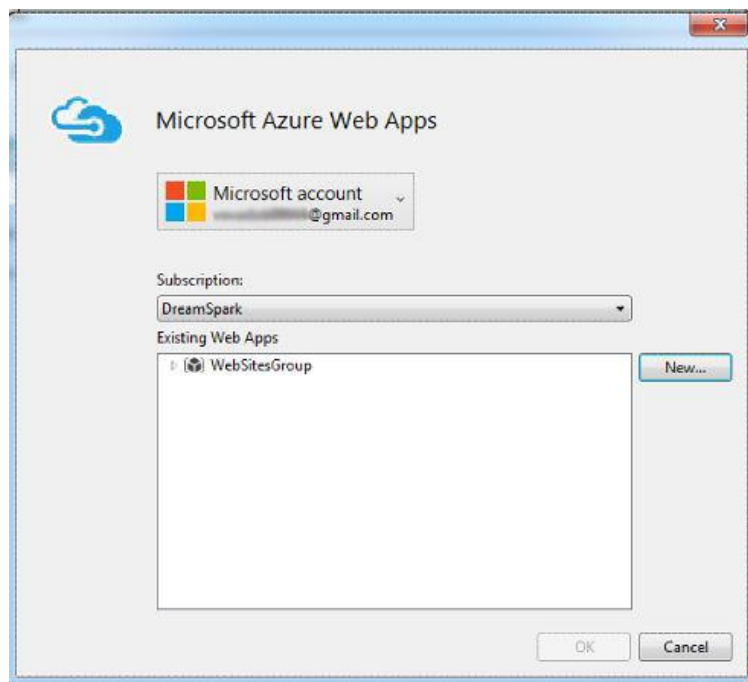
Нажимаємо праву кнопку миші на проект і вибираємо **Publish...**



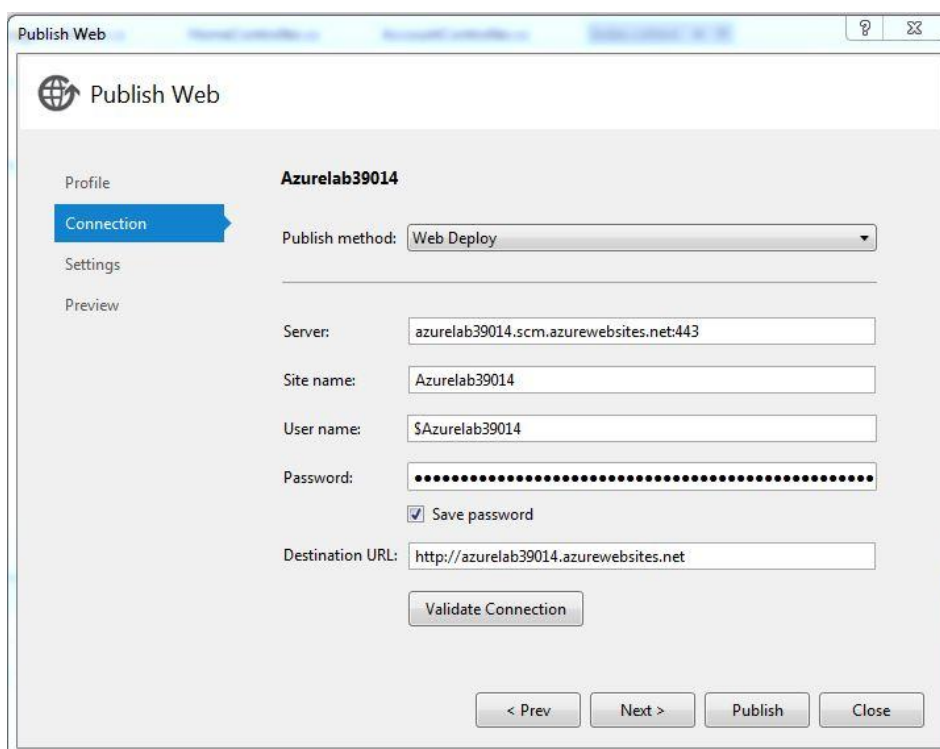
Вибираємо метод публікації в даному випадку **MS Azure Web Apps** і нажимаємо на кнопку **Next**.



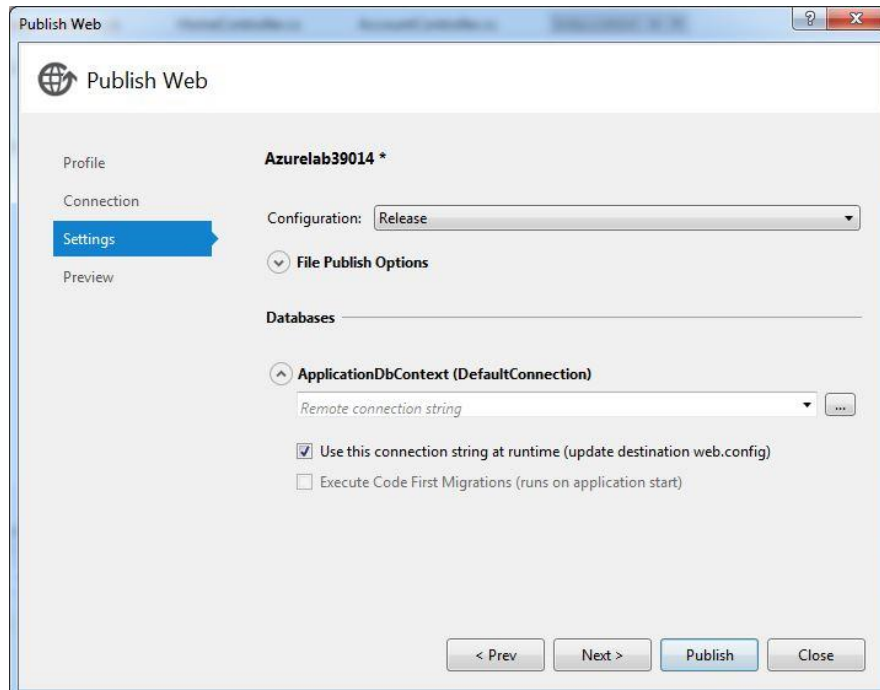
Вибираємо **Existing Web Apps** з списку і нажимаємо **OK** (вона вже створена у вікні конфігурації MS Azure Web App) Підписка (**Subscription**) є тільки від DreamSpark.



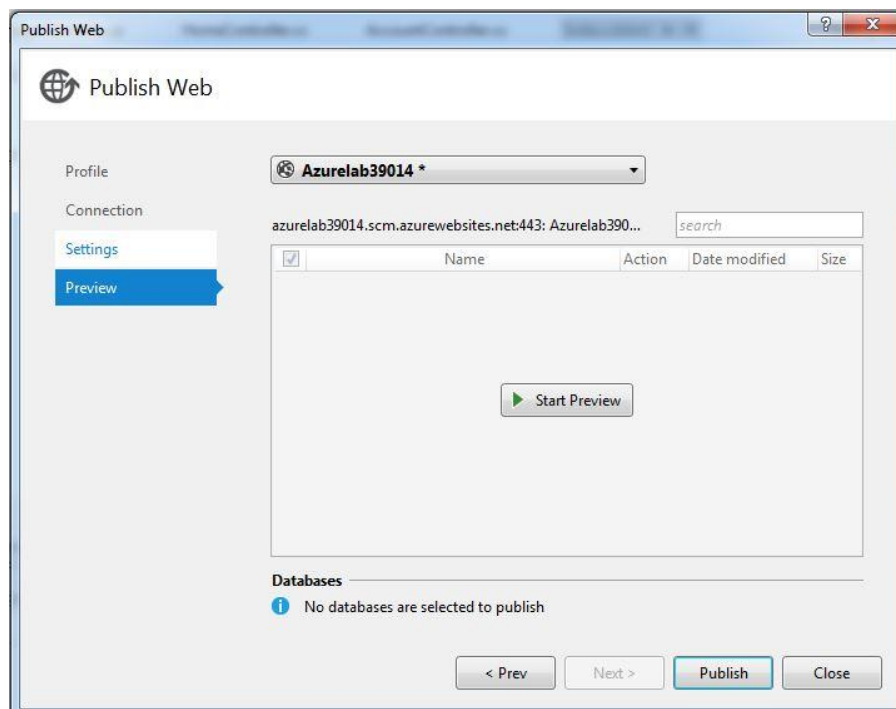
Після заповнення пункту **Profile**, переходимо до пункту **Connection** в якому система сама перевіряє усі дані введені на попередніх етапах. І заповнює поля, для перевірки коректності адреси ресурсу, створена кнопка **Validate Connection**. Після перевірки нажимаємо кнопку **Next**



Налаштовуємо даний сайт **Configuration→Release**, також можна додати **Databases**, ще встановлено автоматичне оновлення сайту через **web.config**



Наступним кроком є перевірка даних, які будуть надіслані на Host.



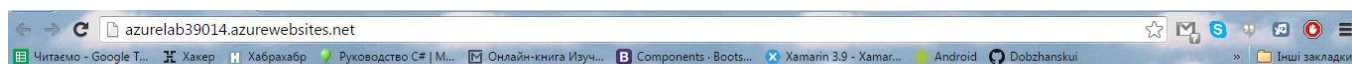
Далі нажимаємо **Publish** і даний сайт компілюється і розміщується на хостингу Azure.

```
Output
Show output from: Build
1>----- Build started: Project: Azure_lab_3, Configuration: Release Any CPU -----
1> Azure_lab_3 -> C:\Users\Vova\Desktop\TestWork\Azure_lab_3\Azure_lab_3\bin\Azure_lab_3.dll
2>----- Publish started: Project: Azure_lab_3, Configuration: Release Any CPU -----
2>Transformed Web.config using C:\Users\Vova\Desktop\TestWork\Azure_lab_3\Azure_lab_3\Web.Release.config into obj\Release\TransformWebConfig\transformed\Web.config.
2>Auto ConnectionString Transformed obj\Release\TransformWebConfig\transformed\Web.config into obj\Release\CSAutoParameterize\transformed\Web.config.
2>Copying all files to temporary location below for package/publish:
2>obj\Release\Package\PackageTmp.
2>Start Web Deploy Publish the Application/package to https://azurelab39014.scm.azurewebsites.net/msdeploy.axd?site=Azurelab39014 ...
2>Adding ACL's for path (Azurelab39014)
2>Adding ACL's for path (Azurelab39014)
2>Updating file (Azurelab39014\Views\Web.config).
2>Updating file (Azurelab39014\Web.config).
2>Adding ACL's for path (Azurelab39014)
2>Adding ACL's for path (Azurelab39014)
2>Publish Succeeded.
2>Web App was published successfully http://azurelab39014.azurewebsites.net/
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
===== Publish: 1 succeeded, 0 failed, 0 skipped =====
```

Після компіляції повинно обов'язково відобразити два поля, які підтвердять компіляцію успішною.

```
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
===== Publish: 1 succeeded, 0 failed, 0 skipped =====
```

Після успішної компіляції відобразиться сайт розміщений на хостингу Azure.



This is three laboratory Azure

© 2015 - Three Application

Завдання

1. Парні варіанти мають використати шаблон **Web Forms**. Сайт повинен запускатися через аккаунт MS Azure і відображувати:

- прізвище ім'я по батькові;
- номер групи;
- дату виконання завдання.

2. Непарні варіанти мають використати шаблон **Web API**. Сайт повинен запускатися через аккаунт MS Azure і відобразити **КРКН 15.261.06.01 ПЗ** :

- вказуючи замість КРКН назву предмету;
- рік, номер залікової книги, варіант по списку, номер лабораторної.

Запитання до лабораторної

- 1) Назвіть рівні архітектури хмарного обчислювального середовища.
- 2) Які найпопулярніші інфраструктури сучасних Cloud-систем?
- 3) Чим хмарний сервіс Amazon EC2 відрізняється від Google App Engine?
- 4) Чи має сервіс Microsoft Windows Azure Platform значні переваги перед Amazon EC2?
- 5) З якими проблемами стикаються користувачі хмарних систем?

Список використаної літератури:

3. Qi Zhang. Cloud computing: state-of-the-art and research challenges / Zhang Q., Cheng L., Boutaba R.: J Internet Serv Appl (2010).
4. Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency. / Calder B., Wang J., Ogus A., Nilakantan A., Skjolsvold A. : SOSPP '11 Proceedings of the Twenty-Third ACM Symposium on Operating.
5. XenSource Inc [Електронний ресурс]: Xen – Режим доступу: <http://www.xensource.com/>.

ДЛЯ НОТАТОК

[illegible]

ДЛЯ НОТАТОК

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Навчально-методична література

Шимчук Г.В., Маєвський О.В.,
Назаревич О.Б., Небесний Р.М.

**Методичні вказівки до виконання
лабораторних робіт
з дисципліни**

**“ГРІД-СИСТЕМИ ТА ТЕХНОЛОГІЇ ХМАРНИХ
ОБЧИСЛЕНЬ”**

для студентів освітніх рівнів «спеціаліст», «магістр»
122 «Комп’ютерні науки та інформаційні технології»

Комп’ютерне верстання *А.П. Катрич*

Формат 60х90/16. Ум. друк. арк. 1,35. Тираж 10 пр. Зам. № 2819.

Тернопільський національний технічний університет імені Івана Пулюя.

46001, м. Тернопіль, вул. Руська, 56.

Свідоцтво суб’єкта видавничої справи ДК № 4226 від 08.12.11.